Digitaler
Kleinrechen-
automat

C 8205

Programmierung und Bedienung

vvb maschinelles rechnen
Zentralstelle für Bildung

VVB Maschinelles Rechnen
Zentralstelle für Bildung
102 Berlin
Postfach 19

Autoren: Wolfgang Apitz, Zentrum Organisation und Datenverarbeitung, Berlin

Karlheinz Burwieck, VEB Maschinelles Rechnen, Berlin

Druckgenehmigungs-Nr.: (588)Ag 710/113/74
Alle Rechte vorbehalten
Druck: VEB Kleinoffsetdruckerei
108 Berlin
Buchbinderische Verarbeitung: VEB Berliner Buchgewerbe
102 Berlin

Inhalt

		Seite
0.	Vorbemerkungen	7
1.	Funktion und Aufbau des C 8205	7
1.1.	Technische Daten	7
1.2.	Aufbau	8
1.2.1.	Geräte der 1. Peripherie	8
1.2.2.	Zentraleinheit	8
1.2.2.1.	Speicher	10
1.2.2.2.	Ein- und Ausgabewerk	10
1.2.2.3.	Akkumulator und Register	11
1.3.	Informationsdarstellung	11
1.3.1.	Triadendarstellung	11
1.3.2.	Befehlswort	12
1.3.3.	Festkommadarstellung	14
1.3.4.	Gleitkommadarstellung	15
1.4.	Befehlsschleife des C 8205	16
1.4.1.	Die 4 Phasen der Befehlsabarbeitung	17
2.	Die Programmierung des C 8205 in der Maschinensprache	20
2.1.	Besonderheiten bei der Programmierung in der Maschinensprache	20
2.2.	Beschreibung des Befehlssystems	20
2.2.1.	Befehlstyp	20
2.2.2.	Operationsart	22
2.2.3.	Operationszusatz	22
2.2.4.	Einzelbefehle	23
2.2.4.1.	Additionsbefehle	23
2.2.4.2.	Transportbefehle	24
2.2.4.3.	Subtraktionsbefehle	25
2.2.4.4.	Konjunktion	26
2.2.4.5.	Vorzeichenbefehle	26
2.2.4.6.	Adressenlosebefehl	28
2.2.5.	Organisationsbefehle 1. Art	29
2.2.5.1.	Unbedingter Sprung	29
2.2.5.2.	Sprünge gemäß AC-Inhalt	30
2.2.5.3.	Unbedingter Stopp	30

		Seite
2.2.5.4.	Sprung mit Rückkehrabsicht	31
2.2.5.5.	Befehle der Ein- und Ausgabe	32
2.2.6.	Markierte Einzelbefehle B = 3	34
2.2.7.	Organisationsbefehle 2. Art	34
2.2.7.1.	Vergleichersprünge	35
2.2.7.2.	Weitere Organisationsbefehle 2. Art	36
2.2.8.	Gruppen- und Wiederholungsbefehle	37
2.3.	Programmierbeispiele	38
3.	Die Programmierung des C 8205 mit Hilfe von Interpretiersystemen	43
3.1.	Die Notwendigkeit der Anwendung von Interpretiersystemen zur rationellen Programmierung des Rechners	43
3.2.	Übersicht über vorhandene Interpretiersysteme	44
3.3.	Das Bausteininterpretiersystem (BIS)	44
3.3.1.	Beschreibung der Bausteine	45
3.3.1.1.	Das kleine Eingabeprogramm (KEP)	45
3.3.1.2.	Das Minimalprogramm	46
3.3.1.3.	Eingabebausteine	51
3.3.1.4.	Ausgabebausteine	55
3.3.1.5.	Rechenbausteine	58
3.3.1.6.	Serviceprogramme	61
3.4.	Darstellung der Zusammenhänge zwischen dem KEP, dem Minimalprogramm, den Bausteinen und den Anwenderprogrammen	63
3.4.1.	Funktionsmechanismus	63
3.4.2.	Anschlußbedingungen zu den Anwenderprogrammen - Register, Anschluß- und Ansprungzellen, Kommunikation -	66
3.5.	Programmierbeispiele	67
4.	Programmiersystem mit symbolischer Adressierung (PROSA)	71
4.1.	Hinweise für die Anwendung der Assemblersprache PROSA	71
4.2.	Format der Assemblersprache PROSA	72
4.2.1.	Befehle	72

		Seite
4.2.1.1.	Der Name	73
4.2.1.2.	Die Operation	74
4.2.1.3.	Der Operand	75
4.2.1.4.	Das Kennzeichen	77
4.2.1.5.	Das Übersetzungsverhältnis	77
4.2.2.	Bereichsdefinitionen	79
4.2.3.	Konstantendefinitionen	80
4.2.4.	Zeichenketten	81
4.3.	Herstellung der Datenträger	82
4.3.1.	Quellenprogrammstreifen	82
4.3.2.	Datenstreifen	83
4.4.	Korrekturmöglichkeiten bei fehlerhaften Programmen	84
4.4.1.	Streichen falscher Instruktionen	84
4.4.2.	Einfügen fehlender Instruktionen	85
4.4.3.	Korrigieren fehlerhafter Instruktionen	86
4.4.4.	Beispiele für die Korrektur	87
4.5.	Regieanweisung für den Assembler	89
4.5.1.	Speicherplatzbelegung und Hinweise für die Abspeicherung	89
4.5.2.	Der erste Übersetzungslauf	90
4.5.3.	Der zweite Übersetzungslauf	91
4.5.4.	Beispiel für eine vollständige Übersetzung	92
4.5.5.	Fehlerausschriften bei der Übersetzung	95
4.6.	Befehlsliste	97
4.7.	Programmierbeispiele mit Übersetzungsprotokollen	102
5.	Die Bedienung des C 8205	111
5.1.	Die Geräte der 1. Peripherie	111
5.1.1.	Der Lochbandleser 303 bzw. FS 750	111
5.1.2.	Der Lochbandstanzer	113
5.1.3.	Das Schreibwerk	113
5.1.4.	Tastatur und Anzeigeeinheit	115
5.1.4.1.	Tastatur	115
5.1.4.2.	Lampenfeld	119
5.2.	Inbetriebnahme des Rechners	121

		Seite
5.3.	Eingabe des Systems	121
5.3.1.	Eingabe des Minimalprogramms	121
5.3.2.	Eingabe der Bausteine	122
5.3.3.	Eingabe von Programmen (Anwenderprogramme)	122
5.4.	Eingabe über Tastatur bzw. über Schreibwerktastatur	122
5.5.	Herstellung der Datenträger	124
5.5.1.	Programmstreifen	125
5.5.2.	Datenstreifen	125
5.5.3.	Korrekturmöglichkeiten bei der Lochbandherstellung	125
5.6.	Korrektur des Inhalts von Speicherplätzen	127
5.7.	Programmstart	127
5.8.	Eingriffsmöglichkeiten beim Auftreten bestimmter Betriebszustände bei der Abarbeitung von Programmen	128
5.9.	Hinweise zur rationellen Testung von Programmen	129
5.9.1.	Vorbereitung des Programmtests durch die Trockentests	130
5.9.2.	Festlegung des Testablaufs	133
5.9.3.	Zusammensetzen der Bausteine	134
5.9.4.	Adressierungsregeln für die Abspeicherung von Bausteinen	135
5.9.5.	Serviceprogramme als Testhilfen	136
5.9.5.1.	Bereichsausschreiben	136
5.9.5.2.	Protokollprogramm	140
5.9.5.3.	Druck von < 7777 >	148
5.9.5.4.	Vergleichsprogramm	148
5.9.5.5.	Bereichsausstanzen	149
	Anschlußzellender Bausteine des BIS	151
	Zusammenstellung der häufigsten Operationen	152
	Eingabekennzeichen	155
	Befehlsfolge zum Duplizierprogramm (R 300-Code)	156
	Codierung R 300	157
	Literaturnachweis	159

0. Vorbemerkungen

Der programmgesteuerte digitale Kleinrechenautomat C 8205, wie er seit 1969 vom VEB Rechenelektronik Meiningen gebaut wird, stellt sowohl die konsequente Umsetzung der Erfahrungen beim Bau und bei der Anwendung des SER 2 als auch die Verwirklichung der Anwenderforderungen nach größerer Speicherkapazität und Flexibilität dar. Der C 8205 gehört zu den Rechenanlagen der 2. Rechnergeneration. Durch seinen logischen Aufbau, sein analytisches flexibles Befehlssystem sowie die variablen Möglichkeiten der Programmierung ist er für die verschiedensten Aufgaben der maschinellen Datenverarbeitung einsetzbar. Er kann ebenso zur Lösung wissenschaftlich-technischer wie mathematisch-ökonomischer Probleme Anwendung finden. Die variablen Möglichkeiten der Programmierung von der Maschinensprache bis zur symbolischen Adressierung in verschiedenen Systemen zeigen Analogien zur Arbeitsweise von Rechenanlagen der 3. Generation. Diese Vorzüge und der verstärkte Einsatz des C 8205 in unserer Volkswirtschaft lassen ihn für die Grundlagenbildung im Rahmen der Ausbildung von Facharbeitern für Datenverarbeitung besonders geeignet erscheinen.

1. Funktion und Aufbau des C 8205

1.1. Technische Daten

Der C 8205 ist ein Einadreßrechner mit fester Wortlänge. Die Abarbeitung der Befehle erfolgt seriell, d.h. in logischer Reihenfolge der Abspeicherung nacheinander. Speichermedium ist eine Magnettrommel mit einer Umdrehungszahl von 18000 U/min.
Die gesamte Speicherkapazität beträgt 4096 Worte (à 33 bit). Es können maximal 2000 verdrahtete Operationen pro Sekunde ausgeführt werden.
Die Zeit für die Ausführung einer Operation liegt nach Art des Befehls zwischen 0,1 und 0,5 ms. Die Taktfrequenz von 316 kHz, die u.a. die Arbeitsgeschwindigkeit des Rechners bestimmt, sorgt für das synchrone Arbeiten aller

Baugruppen. Sie ergibt sich aus der Umdrehungszahl der Trommel und der Schreibdichte von 32 Wörtern (à 33 bit) je Spur.

1.2. Aufbau

1.2.1. Geräte der 1. Peripherie

Zu den Geräten der 1. Peripherie gehören:

2 Lochbandleser (elektromechanisch bzw. fotoelektrisch)
1 alphanumerisches, in allen Funktionen ansteuerbares Schreibwerk
1 Lochbandstanzer

Die maximal erreichbare Arbeitsgeschwindigkeit der peripheren Geräte wird durch die technischen Daten des jeweiligen Einzelgerätes und durch das Programm bestimmt.

Der Lochbandleser kann Geschwindigkeiten von maximal 130 ... 150 Z/s erreichen. Es können 8-Kanal- bzw. 5-Kanal-Lochbänder gelesen werden.

Das alphanumerisch ansteuerbare Schreibwerk mit Ein- und Ausgabefunktion hat eine Ausgabegeschwindigkeit von maximal 10 Z/s (theoretischer Wert).

Der Lochbandstanzer kann maximal 50 Z/s auf Lochband ausgeben. Die Ausgabe ist in einem beliebigen Code möglich und wird durch das jeweilige Interpretiersystem bzw. Programm bestimmt.

1.2.2. Zentraleinheit

Vgl. Abb. 1 Blockschaltbild des C 8205 (vereinfacht)

Blockschaltbild des C 8205 (vereinfacht)

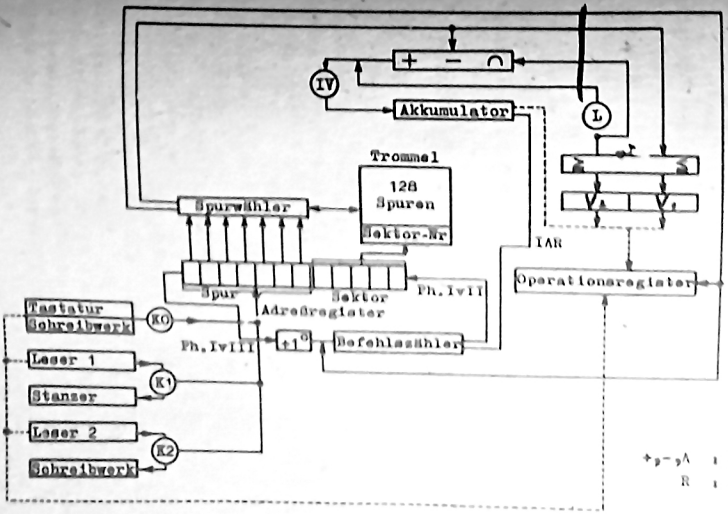


Abb. 1

$+_p - _pA$: Rechenwerk

R : R - Befehl (z.B. Adressenlese-befehl)

V_1, V_2 : Vergleicher

1.2.2.1. Speicher

Als Speicher wird beim C 8205 eine Magnettrommel verwendet. Auf 128 Spuren sind 32 Zellen à 33 bit untergebracht. Mit einigen Einschränkungen sind alle Speicherzellen gleichermaßen zur Speicherung von Befehlen, Konstanten und Variablen geeignet. Eine Trennung von Befehlen und Konstanten bzw. Variablen (wie beim SER 2d) ist nicht vorgesehen.
Die Gesamtspeicherkapazität der Trommel beträgt somit 4096_{10} Speicherplätze. Für die Speicherung von Daten und Befehlen stehen, je nach Interpretiersystem, ca. 2500...3000 Plätze zur Verfügung.

1.2.2.2. Ein- und Ausgabewerk

Die Eingabe kann wahlweise über 3 Kanäle und die Ausgabe über 2 Kanäle erfolgen. Die Belegung der Kanäle ist wie folgt realisiert:

K0 Tastatur und Schreibwerktastatur (bei Eingabe)
K1 Leser 1 bzw. Lochbandstanzer
K2 Leser 2 bzw. Schreibwerk
K3 Reserve (z.Z. nicht belegt)

Zur Realisierung eines Ausgleichs zwischen der Rechengeschwindigkeit der Zentraleinheit und der mechanisch bedingten Trägheit der peripheren Geräte erfolgt die Eingabe gepuffert (einzeichengepuffert).
Der Rechner ist so aufgebaut, daß mit einer Operation jeweils nur ein Zeichen über Puffer und Adreßregister in den Akkumulator gelesen werden kann. Eine Eingabe von mehrstelligen Zahlen oder Alphazeichen kann ökonomisch nur durch die Verwendung von Unterprogrammen realisiert werden. (Die einzelnen Interpretiersysteme enthalten diese Routinen.) Die reale Lesegeschwindigkeit der Lochbandleser wird vom verwendeten Eingabeprogramm bestimmt. Taktimpulse der Zentraleinheit steuern dabei die Leser, die nach einem Start-Stopp-Prinzip arbeiten.

1.2.2.3. Akkumulator und Register

Der Rechner verfügt über ein Rechenregister (Akkumulator, nachfolgend AC genannt), das, wie auch der Befehlszähler, als sogenanntes Umlaufregister mit Hilfe der Trommel realisiert ist.
Durch das Rechenwerk werden die Grundoperationen (Addition, Subtraktion und die logischen Operationen(+, -, ∩) ausgeführt. Der AC dient dabei, wie bei Einadreßrechnern allgemein üblich, als Arbeitszwischenspeicher für die Speicherung eines Operanden. Die Ergebnisbildung erfolgt bei den verdrahteten Grundoperationen ebenfalls im AC.

Für die zwischenzeitliche Speicherung eines Befehls zur Abarbeitung dienen ferner das Operations- und das Adreßregister, die zusammen das Befehlsregister bilden. Drei mögliche Vergleichszustände (aus dem Vergleich des Inhalts einer Speicherzelle mit dem AC-Inhalt) können in einem Flip-Flop-Paar (Vergleicherzustand 1, 2 und 3) gespeichert werden.

1.3. Informationsdarstellung

Das Maschinenwort des C 8205 hat eine konstante Länge von 33 Bit. Die interne Darstellung von Informationen ist grundsätzlich dual. Alle Speicherplätze können für die Speicherung von Befehlen, Konstanten und Variablen verwendet werden. Die Interpretation erfolgt durch das Verarbeitungsprogramm. Der Inhalt einer Speicherzelle kann dabei als Befehlswort bzw. als Zahlwort interpretiert werden. Grundsätzlich können mit dem Inhalt aller Zellen (Konstanten, Befehle, variable Daten) die gleichen Operationen ausgeführt werden.

1.3.1. Triadendarstellung

Um eine ökonomische Form der Notierung von Zahlen zur Eingabe und Abspeicherung zu erhalten, werden je 3 bit zu einer Oktalzahl zusammengefaßt. Bei dieser Form der

externen Darstellung ist es möglich, wenn die bits einer Triade eine Wertigkeit von 4, 2 und 1 haben, Zahlen von 0...7 darzustellen. Die 33 Bit des Maschinenwortes ergeben demzufolge 11 Triaden (Oktaldarstellung).

Vgl. Abb. 2 und 3, Seite 13

1.3.2. Befehlswort

Durch das Befehlswort (Maschinenbefehl) des C 8205 werden von den 33 Bit im allgemeinen nur 21 Bit belegt. Es werden dabei 12 Bit zur Darstellung der Adresse und 9 Bit zur Darstellung des Operationsteils benötigt. Extern wird der Befehl als Oktalzahl notiert.

Externe Darstellung: $a_1\ a_2\ a_3\ a_4$ BAZ.

Vgl. Abb. 4 (interne Darstellung), Seite 13

Dabei bedeuten: $a_1\ a_2\ a_3\ a_4$	Adreßteil
B	Befehlstyp
A	Operationsart
Z	Operationszusatz

Der Adreßteil setzt sich aus Spur- und Sektornummer zusammen. Der Einfachheit halber werden Spur- und Sektornummer zu einer vierstelligen Oktalzahl zusammengefaßt. Daraus ergeben sich bei 4096_{10} Speicherplätzen Adressen von 0000_8.....7777_8.

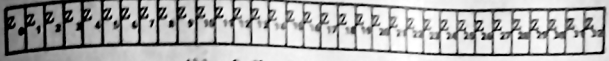


Abb. 2 Maschinenwort

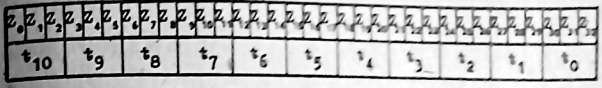


Abb. 3 Triadendarstellung

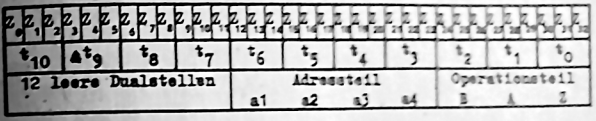


Abb. 4 Befehlswort

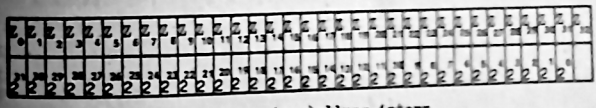


Abb. 5 Festkommadarstellung intern

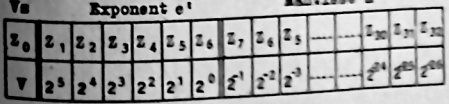


Abb. 6 Gleitkommadarstellung intern

1.3.3. Festkommadarstellung

Bei der Festkommadarstellung wird jeder Dualstelle ein fester Stellenwert zugeordnet.
Interne Darstellung:

Vgl. Abb. 5, Seite 13

Da die Dualstelle z_{32} für Rundungszwecke benutzt wird, ist die Stelle mit dem höchsten Stellenwert $z_o = 2^{31}$. Demzufolge ergibt sich der Zahlenbereich

$$z = 2^{31} - 1 = 2\ 147\ 483\ 647.$$

Die kleinste von Null verschiedene Zahl ist demnach $2^o = 1$. Negative Zahlen werden als Zweierkomplement dargestellt (B-Komplement).
Man sollte sich bei der Festkommadarstellung das Komma grundsätzlich außerhalb des Maschinenwortes denken. Die gespeicherte Zahl ist dabei eine ganze Zahl ohne Komma.

Externe Darstellung:
In der Festkommadarstellung ist die Stellenzahl einer Dezimalzahl auf maximal 10 Dezimalziffern begrenzt.

$$d = (-)\ d_9\ d_8\ d_7\ d_6\ d_5\ d_4\ d_3\ d_2\ d_1\ d_o.$$

Als größte abzuspeichernde Zahl ergibt sich:

$$d = 2^{31} - 1 = 2\ 147\ 483\ 647.$$

Beispiel:

Bitposition	z_o	z_1	...	z_{28}	z_{29}	z_{30}	z_{31}	z_{32}
intern	0	0	...	1	1	0	1	0
extern							1	3

Das Bit z_{32} bleibt bei der Festkommadarstellung für Rundungszwecke frei.

1.3.4. Gleitkommadarstellung

In der Gleitkommadarstellung wird eine Zahl durch Mantisse und Exponent dargestellt. Dabei werden Mantisse, Exponent und Vorzeichen der Zahl in einem Maschinenwort gespeichert. z_0 des Maschinenwortes enthält das Vorzeichen, von $z_1 \ldots z_6$ ist der Exponent und von $z_7 \ldots z_{32}$ ist die Mantisse gespeichert.
Interne Darstellung:

$$z = (-)\ m \cdot 2^e$$

Mathematisch betrachtet wäre zur Darstellung einer Zahl kleiner als Null ein negativer Exponent erforderlich. Zur Vermeidung komplizierter Vorzeichenrechnungen mit dem Exponenten wird dieser nach der Vorschrift $\acute{e} = e + 32$ maschinenintern erhöht.

Vgl. Abb. 6, Seite 13

Die externe Darstellung ist auf zweierlei Art möglich. Die Zahl kann in Dezimalschreibweise bzw. in Gleitkommadarstellung notiert werden.

An dieser Stelle sei nochmals darauf hingewiesen, daß sowohl die Ein- als auch die Ausgabe von Zahlen sowie die Interpretation des Inhalts der Maschinenworte durch das jeweils zur Anwendung kommende Interpretiersystem, das die entsprechenden Routinen enthalten muß, bestimmt werden.

1.4. Befehlsschleife des C 8205

Vgl. Abb. 7, Seite 16

Befehlsschleife des C 8205

Schema der Befehlsabwicklung

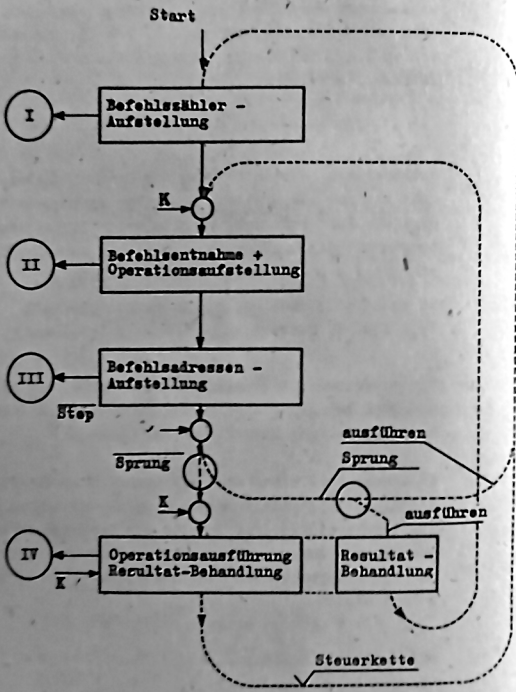


Abb. 7

AG

1.4.1. Die 4 Phasen der Befehlsabarbeitung

Die Abarbeitung von Maschinenbefehlen erfolgt nach einem bestimmten logischen Algorithmus (Befehlsschleife). Sie verläuft in 4 Phasen. Der Rechner braucht dabei für den Durchlauf einer Phase im allgemeinen eine Wortzeit von ca. 0,1 ms. Bei Wiederholungs- und Gruppenbefehlen vervielfacht sich die Zeit beim Durchlaufen der Phase . Außerdem tritt ein zusätzlicher Zeitbedarf für die Herstellung der Übereinstimmung der eingestellten Adresse mit der Adresse auf der Trommel auf (Koinzidenz).
Je nach Befehlstyp werden alle Phasen der Befehlsschleife durchlaufen, oder der Ablauf wird entsprechend modifiziert.

In der Phase I wird der Inhalt des Befehlszählers in das Adreßregister transportiert. Der vorhergehende Inhalt des Adreßregisters wird um eine Adresseneinheit erhöht und in den Befehlszähler gebracht. Tatsächlich wird also in Phase I der Inhalt des Befehlszählers um eine Adresseneinheit erhöht und die äußere Adresse (= lfd. Adresse der Abspeicherung des abzuarbeitenden Befehls) im Adreßregister zur Verfügung gestellt.

Vor Beginn der Phase II kann eine Wartezeit auftreten bis zur Übereinstimmung der im Adreßregister eingestellten Adresse (äußere Adresse) und der erforderlichen Trommelstellung. Diese Zeit kann maximal der Zeit für eine Trommelumdrehung entsprechen. Bei Erreichen der Übereinstimmung wird der Inhalt der entsprechenden Speicherzelle gelesen und als Befehl interpretiert. Der aus 9 Dualstellen (3 Triaden = BAZ) bestehende Operationsteil wird in das Operationsregister transportiert. Der Adreßteil des Befehls wird im Befehlszähler zwischengespeichert. Bei sogenannten R-Befehlen (wird an späterer Stelle ausführlich besprochen) erfolgt eine Umstellung des Operationsteiles. Bedingte Sprünge werden in dieser Phase bei erfüllter Sprung-

bedingung in unbedingte Sprünge umgestellt.

In Phase III wird der im Befehlszähler zwischengespeicherte Adreßteil a in das Adreßregister gebracht. Der um 1^{0} erhöhte Adreßregisterinhalt läuft in den Befehlszähler (äußere Adresse) ein. Es erfolgt ferner die Abfrage auf Stoppbefehl. Bei bedingten Sprungbefehlen mit nicht erfüllter Sprungbedingung wird die Arbeit nach einer Wartezeit bei Phase I erneut begonnen, d.h., diese Befehle wirken wie Befehle ohne Operation (Leerbefehl). Eine eventuell geforderte Löschung des AC-Inhalts erfolgt ebenfalls in Phase III vor Ausführung der Operation.

Mit Ausnahme der angeführten Sonderfälle schließt sich nach Ablauf einer Wartezeit bis zur Erreichung der Koinzidenz (Übereinstimmung von Operandenadresse - innerer Adresse - und der Trommelstellung) Phase IV an. Entsprechend der im Operationsregister stehenden geforderten Operation wird jetzt der Inhalt des AC mit dem Inhalt der im Adreßteil angegebenen Speicherzelle verknüpft.
Bei markierten Einzelbefehlen wird zusätzlich einer der drei möglichen Vergleichszustände (aus dem Vergleich von AC-Inhalt mit dem Inhalt der Adresse oder dem Vergleich des AC-Inhalts mit Ø) gespeichert.
Eine möglicherweise geforderte Resultatbehandlung (Rechts- bzw. Linksverschiebung) mit dem AC-Inhalt wird ausgeführt. Je nach Befehlstyp kann diese Phase 1 bis 32 mal ausgeführt werden.
Der Durchlauf wird bei Phase I fortgesetzt.

Beispiel:

lfd. Adresse der Abspeicherung (äußere Adresse)	Adresse der Zelle, mit deren Inhalt die Operation ausgeführt wird (innere Adresse)	BAZ-Operationsteil
2 1 0 1 = b	4 7 3 0 = a	215 { in Phase III AC löschen; in Phase IV Ergebnis 1 Bit nach links verschieben; Addition; Einzelbefehl

Phase I

⟨AR⟩ : = ⟨BZ⟩ : = 2100
⟨BZ⟩ : = ⟨AR⟩ : = 2100 + 1^{0} : = 2101

Phase II

- Der Inhalt von b = 2101 wird gelesen und als Befehl interpretiert.
- Operationsteil und Adreßteil werden getrennt. Der Operationsteil läuft in das Operationsregister ein, der Adreßteil wird im Befehlszähler zwischengespeichert.

⟨OR⟩ : = Operationsteil
⟨BZ⟩ : = Adreßteil

Phase III

Gemäß Z = 5 (Operationszusatz) erfolgt Löschung des AC-Inhalts. Außerdem wird der um 1^{0} erhöhte Inhalt des Adreßregisters in den Befehlszähler zurücktransportiert.

⟨AR⟩ : = ⟨BZ⟩
⟨BZ⟩ : = ⟨AR⟩ + 1^{0}

Phase IV

Jetzt wird (gem. A = 2) eine Addition des Inhalts der im Adreßteil bezeichneten Speicherzelle mit dem in Phase III gelöschten AC-Inhalt vorgenommen, d.h., es wird in den AC gelesen (Addition mit vorheriger Löschung = Lesen).
Entsprechend der geforderten Resultatbehandlung (Z = 5) wird das im AC gebildete Ergebnis der Operation um 1 Bit nach links verschoben, das entspricht dual einer Multiplikation mit 2.

Ergebnis: Der Befehl 2101 215 ermöglicht das Lesen des Inhalts einer Speicherzelle bei gleichzeitiger Multiplikation mit 2.

2. Die Programmierung des C 8205 in der Maschinensprache

2.1. Besonderheiten bei der Programmierung in der Maschinensprache

Um einen ökonomischen Aufbau des Rechners bei hoher Leistungsfähigkeit zu erreichen, wurde der Befehlsvorrat des C 8205 auf bestimmte Grundoperationen beschränkt. Durch Variation der vorhandenen Möglichkeiten ergeben sich ca. 370 verschiedene Befehle. Die Multiplikation und Division von Festkommazahlen ist nicht verdrahtet. Für den Programmierer würde das bedeuten, durch sinnvolle Anwendung der Maschinenbefehle über den Umweg von Addition, Subtraktion und bitweiser Verschiebung selbst Routinen (Unterprogramme) aufzubauen, um die nicht verdrahteten Operationen zu realisieren. Der Programmieraufwand wäre dabei unvertretbar hoch. Die Programmierung mit Maschinenbefehlen ist deshalb nur in Kombination mit entsprechenden Unterprogrammen bzw. Interpretiersystemen sinnvoll.

2.2. Beschreibung des Befehlssystems

Der Befehlscode des C 8205 baut auf einem analytischen System auf. Der Operationsteil des Befehls (BAZ) besteht aus drei Oktalzahlen, die in Kombination 372 sinnvolle Operationen mit voneinander verschiedener Wirkung ergeben.

2.2.1. Befehlstyp

B = Ø Organisationsbefehle 1. Art
In Abhängigkeit von der Befehlsart A werden nicht alle Phasen der Befehlsschleife durchlaufen (z.B. bedingte Sprungbefehle).

B = 1 Organisationsbefehle 2. Art
analog Organisationsbefehle 1. Art
(z.B. Sprünge gemäß Vergleicherzustand)

B = 2 Einzelbefehle
Die Operation wird nur mit der im Adreßteil an-

gegebenen Speicherzelle ausgeführt.

B = 3 Markierter Einzelbefehl

Die Wirkung dieses Befehls entspricht der des Einzelbefehls. Zusätzlich wird durch den Vergleich des AC-Inhalts mit dem Inhalt der im Adreßteil angegebenen Speicherzelle ein Vergleichszustand erzeugt und in einem Flip-Flop-Paar gespeichert. Der Vergleich erfolgt vor Ausführung der eigentlichen Operation.

B = 4 Gruppenbefehl (G-Befehl)

Bei diesem Befehlstyp wird die durch A und Z bezeichnete Operation mit allen Zellen einer Spur durchgeführt, d.h., Phase IV der Befehlsschleife wird 32mal durchlaufen (z.B. Löschen aller Zellen einer Spur durch einen Befehl).

B = 5 Verkürzter Gruppenbefehl

Der verkürzte Gruppenbefehl arbeitet wie ein Gruppenbefehl. Es tritt jedoch eine zusätzliche Abbruchbedingung auf. Abbruchkriterium ist das Vorzeichen des AC-Inhalts. Bei negativem AC-Inhalt wird die Operation eine Wortzeit danach beendet.

B = 6 Wiederholungsbefehl

Der Gruppenbefehl beginnt bei der im Adreßteil des Befehls angegebenen Adresse und endet bei der Zelle mit der Adresse xx2). Durch entsprechende Wahl der Adresse kann bestimmt werden, mit welchen Adreßinhalten und wie oft die Operation ausgeführt werden soll. Es kann also eine n-malige Ausführung einer bestimmten Operation erreicht werden: $1 \leq n \leq 32$ (z.B. definierte Anzahl von Additionen mit den Zellen einer Spur bzw. bestimmte Anzahl bitweiser Verschiebungen des AC-Inhalts).

B = 7 Verkürzter Wiederholungsbefehl

Der verkürzte Wiederholungsbefehl arbeitet wie

der Wiederholungsbefehl. Es gilt jedoch auch hier eine zusätzliche Abbruchbedingung analog zum verkürzten Gruppenbefehl.

2.2.2. Operationsart

Die Operationsart A kann nicht unabhängig vom Befehlstyp B betrachtet werden. In Abhängigkeit vom Befehlstyp ergeben sich für die Operation A jeweils verschiedene Wirkungen bei gleicher oktaler Bezeichnung. Z.B. sind 040, 140 und 240 in ihrer Wirkung völlig verschiedene Operationen. Auf die Operationsart A wird deshalb im Zusammenhang mit der Beschreibung ausgewählter Befehle näher eingegangen.

2.2.3. Operationszusatz

Der Operationszusatz kann ebenso wie Befehlstyp und Operationsart die Werte 0 ... 7 annehmen. Er belegt innerhalb des Befehlswortes die Bitpositionen z_{30}, z_{31}, z_{32}. Durch den Operationszusatz werden Resultatbehandlung und eine eventuelle Löschung des AC-Inhalts bestimmt. Dabei steuern z_{30} und z_{31} die Resultatbehandlung und z_{32} die Löschanweisung. Es ergeben sich somit acht voneinander verschiedene Wirkungen.

Z	Wirkung
0	keine Löschung, keine Resultatbehandlung
1	Löschung, keine Resultatbehandlung
2	keine Löschung, Rechtsverschiebung
3	Löschung, Rechtsverschiebung
4	keine Löschung, Linksverschiebung
5	Löschung, Linksverschiebung
6	keine Löschung, zyklische Linksverschiebung
7	Löschung, zyklische Linksverschiebung

Die Löschanweisung bezieht sich immer auf den Inhalt des AC und erfolgt (vgl. Befehlsschleife) vor Ausführung der eigentlichen Operation in der III. Phase der Befehlsabarbeitung.
Der Unterschied zwischen Linksverschiebung und zyklischer Linksverschiebung besteht darin, daß bei der einfachen Linksverschiebung das links überlaufende Bit verlorengeht, während bei der zyklischen Linksverschiebung das links überlaufende Bit in die Stelle z_{32} wieder eingeschrieben wird.

Die Linksverschiebung um 1 Bit entspricht einer Multiplikation mit 2.

Die Rechtsverschiebung um 1 Bit entspricht einer Division durch 2.

Die Resultatbehandlung nach der Operationsausführung in der IV. Phase ist auf den AC-Inhalt bezogen.

2.2.4. Einzelbefehle

Einzelbefehle sind durch B = 2 charakterisiert. Die auszuführende Operation bezieht sich immer nur auf den Inhalt einer Speicherzelle. In der nachfolgenden Darstellung wird nur eine Auswahl der für das Verständnis notwendigen Einzelbefehle näher erläutert. Eine vollständige Zusammenstellung ist im Heft 2 der Reihe "Programmierung und Bedienung des digitalen Kleinrechners C 6205" des VEB Kombinat Zentronik enthalten.

2.2.4.1. Additionsbefehle

Innerhalb der Einzelbefehle sind die Additionsbefehle durch B = 2 und A = 1 charakterisiert. Z ist variabel und kann die oktalen Werte 0 ... 7 annehmen. Es ergeben sich demzufolge acht mögliche Variationen.

z.B. $a_1\ a_2\ a_3\ a_4$ BAZ. BAZ = 210 Einzeladdition

2 1 0

Einzelbefehl Addition keine Löschung, keine Resultatbehandlung

⟨AC⟩ := ⟨AC⟩ + ⟨a⟩

$a_1\ a_2\ a_3\ a_4$ 211.

2 1 1

Einzelbefehl Addition Löschung d. AC vor der Operation, keine Resultatbehandlung

⟨AC⟩ := ⟨a⟩

Der AC wird vor Ausführung der Addition des Inhalts der im Adreßteil des Befehls angegebenen Speicherzelle auf Null gelöscht. Der Befehl wirkt als Legebefehl.

Am Beispiel dieser beiden Additionsbefehle ist deutlich der durch das analytische Befehlssystem gegebene Zusammenhang erkennbar.

$a_1\ a_2\ a_3\ a_4$ 215.

2 1 5

Einzelbefehl Addition Löschung d. AC vor der Operation, Verschiebung des AC-Inhalts um 1 Bit nach links

⟨AC⟩ := 2x ⟨a⟩

Der AC-Inhalt wird vor Ausführung der Operation auf Null gelöscht. Es erfolgt die Addition des Inhalts der im Adreßteil angegebenen Zelle a. Das Resultat wird um 1 Bit nach links verschoben. Das kommt im Dualsystem einer Multiplikation mit 2 gleich.

2.2.4.2. Transportbefehle

Transportbefehle sind durch A = 4 charakterisiert.

Z.B. $a_1\ a_2\ a_3\ a_4$ 240.

2 4 0

Einzelbefehl Transport keine Löschung des AC, keine Resultatbehandlung

Der Inhalt des AC wird unverändert zum Inhalt der im Adreßteil des Befehls angegebenen Speicherzelle transportiert. Der AC-Inhalt wird dabei nicht zerstört.

⟨a⟩ := ⟨AC⟩ Transportbefehl

$a_1\ a_2\ a_3\ a_4$ 241.

2 4 1

Einzelbefehl Transport Löschung d. AC vor der Operation, keine Resultatbehandlung

Der vorher auf Null gelöschte Inhalt des AC wird zum Inhalt der im Adreßteil des Befehls angegebenen Speicherzelle, d.h., der Inhalt der Speicherzelle wird Null.

⟨a⟩ := Ø Löschbefehl

Eine Resultatbehandlung ergibt bei diesem Befehl keinen Sinn, weil dadurch am Ergebnis der Operation keine Veränderung erfolgt.

$a_1\ a_2\ a_3\ a_4$ 242.

2 4 2

Einzelbefehl Transport keine Löschung, Rechtsverschiebung um 1 Bit

Bei diesem Befehl erfolgt anschließend an den Transport eine Verschiebung des AC-Inhalts um 1 Bit nach rechts. Das entspricht im Dualsystem einer Division durch 2.

2.2.4.3. Subtraktionsbefehle

Subtraktionsbefehle sind durch A = 2 charakterisiert.

Z.B. $a_1\ a_2\ a_3\ a_4$ 220.

2 2 0

Einzelbefehl Subtraktion keine Löschung, keine Resultatbehandlung

⟨AC⟩ := ⟨AC⟩ - ⟨a⟩ Subtraktionsbefehl

Vom Inhalt des AC wird der Inhalt der im Adreßteil des Befehls angesprochenen Speicherzelle subtrahiert.

$a_1\ a_2\ a_3\ a_4$ 221.

2 2 1

Einzelbefehl Subtraktion der Inhalt des AC wird vor Operationsausführung gelöscht

⟨AC⟩ := Ø − ⟨a⟩ d.h. ⟨AC⟩ := ⟨−a⟩ Lesebefehl für Komplement

Von dem vorher auf Null gelöschten Inhalt des AC wird der Inhalt der im Adreßteil des Befehls bezeichneten Speicherzelle subtrahiert. Es ergibt sich das Komplement (Negativlesebefehl).

2.2.4.4. Konjunktion

Bei der Konjunktion kann nach zwei prinzipiellen Möglichkeiten unterschieden werden.

A = Ø Konjunktion im AC

A = 5 Konjunktion im Speicher

Bei der Konjunktion im AC wird der Inhalt des AC bitweise mit dem Inhalt der im Adreßteil des Befehls bezeichneten Speicherzelle verknüpft.

⟨AC⟩ := ⟨AC⟩∧⟨a⟩

Z.B. $a_1\ a_2\ a_3\ a_4$ 200.

```
000 000 000 000 000 000 000 111 111 111 111  ⟨AC⟩
000 000 000 000 010 010 010 010 011 000 100  ⟨a⟩
-----------------------------------------------
000 000 000 000 000 000 000 010 011 000 100  ⟨AC⟩ := ⟨AC⟩∧⟨a⟩
```

$a_1\ a_2\ a_3\ a_4$ 250.

Bei der Konjunktion im Speicher steht das Ergebnis der konjunktiven Verknüpfung in der im Adreßteil bezeichneten Speicherzelle. Der Inhalt des AC bleibt dabei unverändert. Die Konjunktion ermöglicht beispielsweise das Ausblenden von Ziffern bzw. Ziffernfolgen mit Hilfe einer Konstanten, die dabei als Maske wirkt.

2.2.4.5. Vorzeichenbefehle sind:

bedingte Addition - Subtraktion A = 3 (außer BAZ = 330, 331)

bedingte Speicherlöschung A = 7

Bei der <u>bedingten Addition - Subtraktion</u> wird allgemein das Bit z_0 als Vorzeichenbit interpretiert. Das entspricht dem Normalfall (außer bei Rechtsverschiebung).

Für $z_0 = 1$ ergibt sich $\langle AC \rangle := \langle AC \rangle + \langle a \rangle$.

Für $z_0 = \emptyset$ ergibt sich $\langle AC \rangle := \langle AC \rangle - \langle a \rangle$.

Das heißt, in Abhängigkeit vom Vorzeichen des AC-Inhalts wird entweder eine Addition oder eine Subtraktion des Inhalts der im Adreßteil des Befehls bezeichneten Speicherzelle mit dem Inhalt des AC durchgeführt.

Z.B. $a_1\ a_2\ a_3\ a_4\ 230$.

```
        100 000 000 000 111 111 111 111 000 000 000
⟨AC⟩    0
⟨a⟩     000 000 000 000 011 010 000 100 010 010 010
        ---------------------------------------------
        1                1 011 010 000 011 010 010 010
⟨AC⟩    00 000 000 00 0 100 101 111 010 101 101 110
```

Für den Fall, daß als Resultatbehandlung eine Rechtsverschiebung vorgesehen ist, wird das Bit z_{32} als Vorzeichenbit interpretiert. Dadurch ergibt sich ein veränderter Ablauf.

Z.B. $a_1\ a_2\ a_3\ a_4\ 232$.

Für $z_{32} = 1$ ergibt sich $\langle AC \rangle := (\langle AC \rangle + \langle a \rangle)\ \backslash 2$.

Für $z_{32} = \emptyset$ ergibt sich $\langle AC \rangle := \langle AC \rangle\ \backslash 2$.

```
                                                  1
⟨AC⟩    000 000 000 000 111 111 111 000 000 000
⟨a⟩     000 000 000 000 011 010 000 010 010 010
        ---------------------------------------------
        1                1 011 001 111 010 010 011
⟨AC⟩    000 000 000 00 0 111 111 111 000 000 000
```

Bei negativem AC-Inhalt wird der Inhalt der im Adreßteil bezeichneten Speicherzelle addiert. Ist der Inhalt des AC positiv, bleibt er unverändert. In beiden Fällen wird anschließend die Resultatbehandlung durchgeführt.

Bedingte Speicherlöschung

Z.B. $a_1\ a_2\ a_3\ a_4$ 270.

Auch bei diesem Befehl hängt der Verlauf von der vorgesehenen Resultatbehandlung ab. Analog zur bedingten Addition gilt auch hier, daß bei einer Rechtsverschiebung z_{32} als Vorzeichenbit interpretiert wird. Im Normalfall gilt z_o als Vorzeichenbit.

Für $z_o = 1$ gilt $\langle AC \rangle := \langle AC \rangle$ und $\langle a \rangle := \emptyset$
Für $z_o = \emptyset$ gilt $\langle AC \rangle := \langle AC \rangle$ und $\langle a \rangle := \langle a \rangle$

Der AC-Inhalt bleibt in beiden Fällen unverändert.

Bei positivem AC-Inhalt wird der Inhalt der im Adreßteil des Befehls bezeichneten Speicherzelle gelöscht.

2.2.4.6. Adressenlesebefehl (R-Befehl)

Dieser Befehl ist durch A = 5 bzw. 7 und Z = 1, 3, 5 gekennzeichnet.

Z.B. $a_1\ a_2\ a_3\ a_4$ 251.

Er ermöglicht das Lesen des Inhalts der Adreßstellen einer im AC stehenden Information.

Die Befehlsschleife läuft folgendermaßen ab:
In Phase II entsteht der Befehlszählerinhalt aus dem Resultat der disjunktiven Verknüpfung der Adressenstellen des $\langle AC \rangle$ mit dem Adreßteil a des aufgerufenen Befehls (R-Befehl).
Im Operationsregister wird z_{27} gelöscht, so daß bei A = 5 die Operation in eine Addition und bei A = 7 in eine bedingte Addition umgestellt wird. Die weitere Abarbeitung erfolgt entsprechend der in Phase II entstandenen Situation.

Beispiel: R-Befehl $b_1\ b_2\ b_3\ b_4$ 251.

AC xxxx $a_1\ a_2\ a_3\ a_4$ xxx.

AR 3000

	I. Phase	II. Phase	III. Phase	IV. Phase
BZ	/	$a_1a_2a_3a_4$	3001	3001
AR	3000	3000	$a_1a_2a_3a_4$	$a_1a_2a_3a_4$
OR	/	211	211	211
AC	BEFEHL	BEFEHL	0	$\langle a_1a_2a_3a_4 \rangle$

Umstellung im Operationsregister:

z_{24}	z_{25}	z_{26}	z_{27}	z_{28}	z_{29}	z_{30}	z_{31}	z_{32}			
O	L	O	L	O	L	O	O	L	=	2 5 1	(R-Befehl)
O	L	O	O	O	L	O	O	L	=	2 1 1	(Addition)

2.2.5. Organisationsbefehle 1. Art

Organisationsbefehle 1. Art sind durch B = Ø charakterisiert. Zu dieser Gruppe von Befehlen gehören u. a. unbedingter Sprung, Sprünge gemäß AC-Inhalt (Positiv-, Negativ- und Überlaufsprung), unbedingter Stopp, bedingter Stopp, Eingabe-Halt-Befehl usw.

2.2.5.1. Unbedingter Sprung

Der unbedingte Sprung ist durch B = Ø und A = Ø gekennzeichnet. Er führt zu einer Veränderung des Befehlszählers. Die normale sequentielle (fortlaufende) Abarbeitung wird unterbrochen und bei der im Adreßteil des Befehls angegebenen Adresse ohne Bedingung fortgesetzt. Entsprechend der Wahl von Z kann der AC-Inhalt einer Resultatbehandlung bzw. Löschanweisung unterworfen werden. Die Normalform des Befehls, wie sie im allgemeinen angewendet wird, stellt sich wie folgt dar:

$$a_1\ a_2\ a_3\ a_4\ 000.$$

2.2.5.2. Sprünge gemäß AC-Inhalt

Negativsprung $a_1\ a_2\ a_3\ a_4$ 040.

Der Ablauf dieser Operation wird durch das Vorzeichen des AC-Inhalts bestimmt. Dabei wird z_o als Vorzeichenbit interpretiert.

$z_o = 1$

AC-Inhalt negativ

Es wird ein Sprung zu der im Adreßteil des Befehls angegebenen Adresse ausgeführt. Das heißt, bei erfüllter Sprungbedingung wird in Phase II der Befehlsschleife dieser Befehl in einen unbedingten Sprung umgestellt.

$z_o = 0$

AC-Inhalt positiv

Bei nichterfüllter Sprungbedingung wirkt dieser Befehl wie ein Leerbefehl, d. h., die Abarbeitung des Programms wird beim nächstfolgenden Befehl in natürlicher Reihenfolge fortgesetzt.

Positivsprung $a_1\ a_2\ a_3\ a_4$ 050.

Die Abarbeitung dieses Befehls wird analog zum Negativsprung durch das Vorzeichen des AC-Inhalts bestimmt. Bei negativem AC-Inhalt (nichterfüllter Sprungbedingung) wirkt dieser Befehl wie ein Leerbefehl. Die Abarbeitung des Programms erfolgt beim nächstfolgenden Befehl in natürlicher Reihenfolge.

2.2.5.3. Unbedingter Stopp

Dieser Befehl führt zu einer Unterbrechung in der Abarbeitung der Befehlsschleife nach Phase III. Der Zustand "Unbedingter Stopp" wird durch die rote Signallampe "US" im Anzeigeteil des Bedienpultes signalisiert. Durch Betätigung der Starttaste "ST" kann die weitere Abarbeitung der Befehlsschleife erreicht wer-

den. Die Programmfortsetzung erfolgt in natürlicher Reihenfolge beim nächstfolgenden Befehl. Durch Betätigung der Tastenkombination H M R kann der unbedingte Stopp in einen Sprungbefehl zu der im Adreßteil des Befehls angegebenen Adresse umgestellt werden.

Z.B. $a_1\ a_2\ a_3\ a_4$ 020.

Wird die o. a. Möglichkeit der Programmfortsetzung nicht benötigt, ist die Angabe einer Adresse bei diesem Befehl nicht erforderlich.

2.2.5.4. Sprung mit Rückkehrbefehl

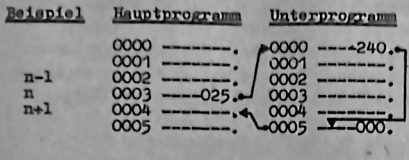
In der Normalform wird dieser Befehl durch A = 2 und Z = 5 charakterisiert.

Z.B. $a_1\ a_2\ a_3\ a_4$ 025.

Mit diesem Befehl werden Unterprogramme erreicht, aus denen unabhängig von ihrer Abspeicherung auf der Trommel eine automatische Rückkehr zum Hauptprogramm erfolgen soll.

Ablauf: Der Befehl wirkt wie ein unbedingter Sprung. Zusätzlich wird der um eine Adresseneinheit erhöhte Befehlszählerinhalt disjunktiv mit dem AC-Inhalt verknüpft. Als Ergebnis dieser Operation steht nach der Sprungausführung die um eine Adresseneinheit erhöhte Absprungadresse im Akkumulator. Das ist die gewünschte Rücksprungadresse nach Abarbeitung des Unterprogramms.

Bedingung: Der AC-Inhalt muß in dem jeweiligen Unterprogramm als erste Operation sichergestellt werden, bevor er durch Operationen des Unterprogramms zerstört wird.



Als erste Operation des Unterprogramms wird durch einen Transportbefehl (hier z.B. 0005 240.) der um 1 Adresseneinheit erhöhte Befehlszählerinhalt - er war durch disjunktive Verknüpfung zum Rücksprungbefehl geworden - zum letzten abzuarbeitenden Befehl des Unterprogramms.

2.2.5.5. Befehle der Ein- und Ausgabe

Eingabe-Halt-Befehl $a_1\ a_2\ a_3\ a_4$ 011.

Auf Grund der Mehrfachausnutzung verschiedener Baugruppen des Rechners wird u. a. das Adreßregister bei der Eingabe mitbenutzt. Der Eingabe-Halt-Befehl bewirkt die Eingabe eines Zeichens über ein vorher zu bestimmendes Eingabegerät (LB-Leser mit Einzeichenpuffer, Schreibwerktastatur oder Tastatur des Bedienfeldes). Jedes eingegebene Zeichen erzeugt im Adreßregister durch den Code und die Zuordnung der 8 Kanäle des Lochbandes zu bestimmten Bitpositionen ein Binärmuster xx. Dieses Binärmuster wird als sogenannte Eingabeadresse verstanden.

Es werden zwei Klassen von Zeichen unterschieden:

- Ist Bit z_{25} mit 1 belegt, wird der Eingabe-Halt-Befehl in einen Additionsbefehl umgestellt.
- Ist Bit $z_{29} = Ø$, wird durch die zweite Klasse von Zeichen der Eingabe-Halt-Befehl in einen Sprungbefehl umgestellt.

Zur praktischen Anwendung bzw. Ausführung des Eingabe-Halt-Befehls muß der so entstandene AC-Inhalt zur Abarbeitung in eine folgende Zelle transportiert werden.

Die Anwendung des Eingabe-Halt-Befehls ist für die praktische Programmierung nur in Sonderfällen von Bedeutung. Zur rationellen Eingabe von Zahlen und Befehlen wurden vom Hersteller geeignete Routinen geschaffen (Eingabebausteine, Duplizierprogramme; vgl. Anlage Duplizierprogramm R 330-Code zur Erläuterung des Eingabe- und Ausgabe-Halt-Befehls).

Ausgabe-Halt-Befehl $a_1\ a_2\ a_3\ a_4$ 030.

Der Ausgabe-Halt-Befehl kann als Funktionssteuerbefehl aufgefaßt werden. Die im Adreßteil des Befehls angegebene sogenannte Ausgabeadresse stellt im eigentlichen Sinne keine echte Adresse dar. Sie ist vielmehr eine Oktalzahl, die ein bestimmtes Binärmuster im Maschinenwort erzeugen soll. Für jede Maschinenfunktion (dazu gehört die Ausgabe aller Zeichen des Schreibwerkes, die Ausgabe von Zeichen über Stanzer, die Anwahl bestimmter Kanäle für Ein- oder Ausgabe usw.) ist durch den Code eine bestimmte Bitkombination erforderlich. Über den Stanzer kann z.B. jede beliebige code-unabhängige Information gestanzt werden. Dabei ergibt sich folgende Zuordnung:

Lochkanal		erforderliche Bitbelegung
1	...	a_{20}
2	...	a_{19}
3	...	a_{18}
4	...	a_{17}
5	...	a_{16}
6	...	a_{15}
7	...	a_{14}
8	...	a_{13}

Einfache Duplizierprogramme bzw. Druckprogramme wandeln den Eingabe-Halt-Befehl durch Addition bzw. Sprung über den Ausgabe-Halt-Befehl um. Auf diese Weise lassen sich mit dem C 8205 auch rationell Umcodierungsprogramme aufbauen.

2.2.6. Markierte Einzelbefehle B = 3

Ob der ⟨ AC ⟩ mit dem Inhalt einer Speicherzelle oder mit Null verglichen wird, richtet sich nach der Operationsart. Bei Operationsarten A ≤ 3 wird der ⟨ AC ⟩ nach der III. Phase mit dem Inhalt der im Adreßteil angegebenen Speicherzelle verglichen.

Bei Operationsarten A ≤ 4 erfolgt der Vergleich der ⟨ AC ⟩ mit Null.

Das Vergleichen geschieht nach folgendem Prinzip:

Am Ende der III. Phase wird generell ein Vergleichszustand 3 erzeugt. Dieser wird in der IV. Phase nach folgender Vorschrift korrigiert:

bei AC < ⟨ a / Ø ⟩ wird zusätzlich Vergleichszustand 1 erzeugt,

bei AC > ⟨ a / Ø ⟩ wird zusätzlich Vergleichszustand 2 erzeugt,

bei AC = ⟨ a / Ø ⟩ bleibt Vergleichszustand 3 erhalten.

Ausnahme: reines Vergleichersetzen.

Z.B. $a_1\ a_2\ a_3\ a_4$ 330.

Bei diesem Befehl wird die Operationsausführung in der IV. Phase unterdrückt. Es erfolgt nur das Erzeugen eines Vergleichszustandes. Der ⟨ AC ⟩ wird mit dem Inhalt der im Befehl angegebenen Adresse verglichen.

Z.B. $a_1\ a_2\ a_3\ a_4$ 331.

Da in der III. Phase der Befehlsabarbeitung der ⟨ AC ⟩ gelöscht wird, erfolgt ein Vergleich von Null mit dem Inhalt der im Befehl angegebenen Speicherzelle.

2.2.7. Organisationsbefehle 2. Art

Diese Gruppe von Befehlen ist durch B = 1 gekennzeichnet. Die Vergleichersprünge (vgl. markierter Einzelbefehl), der Eingabe-Halt-Befehl 2. Art sowie der unbe-

dingte und bedingte Stopp 2. Art sind die wichtigsten Vertreter dieser Gruppe.

2.2.7.1. Vergleichersprünge

Durch Vergleich des AC-Inhalts mit dem Inhalt einer Speicherzelle können drei mögliche Vergleichszustände erzeugt und zwischengespeichert werden. Die Auswertung der Vergleichszustände erfolgt durch Vergleichersprünge.

Vergleichersprung 1 $s_1\ s_2\ s_3\ s_4$ 140.

Ist einer der Vergleicher 1 bzw. 3 gesetzt, so wirkt dieser Befehl wie ein unbedingter Sprung zu der im Adreßteil des Befehls angegebenen Adresse. Sind diese Bedingungen nicht gegeben, so wirkt der Befehl wie ein Leerbefehl, d. h., die Programmabarbeitung wird beim nächstfolgenden Befehl fortgesetzt.

Beispiel:

$\langle AC \rangle < \langle a \rangle$	Sprungbedingung	erfüllt	Vgl. 1 gesetzt
$\langle AC \rangle = \langle a \rangle$	"	"	Vgl. 3 gesetzt
$\langle AC \rangle > \langle a \rangle$	"	nicht erfüllt	Leerbefehl

Vergleichersprung 2 $s_1\ s_2\ s_3\ s_4$ 150.

Ist einer der Vergleicher 2 bzw. 3 gesetzt, so wirkt dieser Befehl wie ein unbedingter Sprung (analog Vergleichersprung 1).

Beispiel:

$\langle AC \rangle > \langle a \rangle$	Sprungbedingung	erfüllt	Vgl. 2 gesetzt
$\langle AC \rangle = \langle a \rangle$	"	"	Vgl. 3 gesetzt
$\langle AC \rangle < \langle a \rangle$	"	nicht erfüllt	Leerbefehl

Vergleichersprung 3 $s_1\ s_2\ s_3\ s_4$ 170.

Ist der Vergleicher 3 gesetzt, wirkt dieser Befehl wie ein unbedingter Sprung (analog Vergleichersprung 1 u. 2).

Beispiel:

$\langle AC \rangle = \langle a \rangle$	Sprungbedingung erfüllt	Vgl. 3 gesetzt
$\langle AC \rangle > \langle a \rangle$	" nicht erfüllt	
$\langle AC \rangle < \langle a \rangle$	" " "	

2.2.7.2. Weitere Organisationsbefehle 2. Art

- Unbedingter Sprung 2. Art
  Die Wirkung entspricht dem unbedingten Sprung 1. Art.

- Eingabe-Halt-Befehl 2. Art
  Der Ablauf dieses Befehls erfolgt analog zum Eingabe-Halt-Befehl 1. Art. Durch B = 1 entsteht bei der Umstellung in Addition eine Operation der Art BAZ = 102, bei der Umstellung in Sprung eine Operation BAZ = 302. Das bedeutet, daß ein zufälliger Vergleichszustand der Vergleicher korrigiert wird.

- Unbedingter Stopp 2. Art
  Dieser Befehl wirkt wie der unbedingte Stopp 1. Art. Analog gilt das über den Eingabe-Halt-Befehl 2. Art Gesagte.

- Bedingter Stopp 2. Art $s_1\, s_2\, s_3\, s_4$ 160.
  Dieser Befehl hängt von der Stellung des Schalters (bedingter Stopp) "BS" ab. Ist der Schalter "BS" gesetzt, geht der Rechner in den Zustand "BS". Dieser Zustand wird durch eine entsprechende Kontrollampe des Anzeigefeldes signalisiert. Durch Betätigung der Taste "ST" bzw. Lösung des Schalters "BS" kann die Unterbrechung aufgehoben werden.
  Zusätzlich wird im Gegensatz zum bedingten Stopp 1. Art bei Betätigung o. g. Tasten der Befehl in BAZ = 36Z umgestellt, d. h., es wird zusätzlich ein Vergleicher gesetzt.

- Leerbefehl $s_1\, s_2\, s_3\, s_4$ 164.
  Die Operationsteile BAZ = 165, 167 bewirken eine Löschung des AC-Inhalts. Die Abarbeitung wird beim

nächstfolgenden Befehl fortgesetzt (keine Operation).

2.2.8. Gruppen- und Wiederholungsbefehle

Gruppenbefehle sind durch B = 4 gekennzeichnet. Ein solcher Befehl beginnt wie ein Einzelbefehl. Die Phasen I...III der Befehlsschleife werden normal analog zum Einzelbefehl durchlaufen. Die Phase IV bezieht sich jedoch auf alle 32 Zellen einer Spur, und die durch 4AZ charakterisierte Operation wird mit den Inhalt aller 32 Zellen einer Spur durchgeführt. Die Reihenfolge der Abarbeitung erfolgt dabei nicht entsprechend der durch die Adressierung gegebenen logischen Reihenfolge, sondern fortlaufend mit der Trommelumdrehung in der Reihenfolge der geometrischen Anordnung der Speicherzellen auf der Trommel.

Z.B. $a_1\ a_2\ a_3\ a_4\ 441$. | Löschen der Inhalte aller Zellen einer Spur

Der Adreßteil des Gruppenbefehls muß eine Adresse aus der jeweiligen Spur darstellen, wobei es nicht unbedingt die Anfangsadresse der Spur sein muß. Trotzdem wird die Operation mit den Inhalten aller Zellen dieser Spur durchgeführt.

Wiederholungsbefehle sind durch B = 6 charakterisiert. Sie wirken ähnlich wie Gruppenbefehle. Der Wiederholungsbefehl beginnt bei einer bestimmten Adresse innerhalb einer Spur und endet bei der Zelle mit der oktalen Bezeichnung (Adresse) xx2) derselben Spur. Durch die Wahl einer bestimmten Anfangsadresse kann die Anzahl der Operationen selbst bestimmt werden! Auch hier erfolgt die Abarbeitung nicht in der Reihenfolge der fortlaufenden Adressierung, sondern in der Reihenfolge der geometrischen Anordnung der Speicherzellen auf der Trommel. Dieser Umstand ist bei der Anwendung der Wiederholungsbefehle unbedingt zu beachten. Die Zuordnung der Anzahl der gewünschten Wiederholungen zu der entsprechenden Adresse wird in der nachfolgenden Tabelle dargestellt. Die Anwendung von Wiederholungsbefehlen

ist notwendig, wenn beispielsweise eine definierte Anzahl von bitweisen einfachen bzw. zyklischen Verschiebungen des AC-Inhalts oder des Inhalts von Speicherzellen benötigt wird. Man realisiert z.B. eine n-malige ($1 \leq n \leq 32$) Verschiebung des AC-Inhalts, indem man sogenannte Scheintransporte durchführt. Der Inhalt des AC wird dabei durch einen Wiederholungstransportbefehl der Gestalt $a_1\ a_2\ a_3\ a_4$ 64Z. (mit Z = 2, 4 oder 6) auf die Zellen einer nicht benötigten oder gesperrten Spur transportiert und dabei gleichzeitig bei jedem Einzeltransport um ein Bit verschoben. Die Anzahl der Verschiebungen wird durch die Wahl der Adresse innerhalb der gewählten Arbeitsspur bestimmt.

Hilfstabelle für die Bestimmung der Adresse bei Wiederholungsbefehlen

n	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
A	00	15	32	07	24	01	16	33	10	25	02	17	34	11	26	03
n	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01
A	20	35	12	27	04	21	36	13	30	05	22	37	14	31	06	23

n = Anzahl der Wiederholungen
A = oktale Adresse des Wiederholungsbefehls in der Spur

Z.B. bedeutet der Befehl xx 27 64Z, daß die Phase IV 13mal durchlaufen wird.

2.3. Programmierbeispiele

Bei den folgenden Beispielen wird angenommen, daß die Programme bereits auf der Trommel abgespeichert sind. Auf Ein- bzw. Ausgabe wurde verzichtet.

Umspeichern des Inhalts von Speicherplätzen

Beispiel 1:

Gegeben sind auf den Adressen 4000 = a , 4001 = b ,

4002 = c. Das Programm beginnt auf der Adresse 4004, die Zelle 4003 wird zur Zwischenspeicherung eines Wertes benutzt.

lfd.Nr.	Adr.-Teil	Op.	Erläuterung	
4000			= a	
4001			= b	
4002			= c	
4003			Zwischenspeicher	
4004	4000	211	⟨AC⟩ := ⟨4000⟩	= a (lesen von a)
4005	4003	240	⟨4003⟩ := ⟨AC⟩	(Transport nach Zwsp.)
4006	4002	211	⟨AC⟩ := ⟨4002⟩	= b (lesen von c)
4007	4000	240	⟨4000⟩ := ⟨AC⟩	(Transp.von c nach 4000)
4010	4003	211	⟨AC⟩ := ⟨4003⟩	(lesen von a)
4011	4002	240	⟨4002⟩ := ⟨AC⟩	= a

Durch Vertauschung von a und c ergibt sich als Ergebnis die Reihenfolge c,b,a.

Beispiel 2: x = 2a - b - c - d

Die Werte a...d stehen auf den Speicherplätzen 4000...4003. Das Ergebnis soll auf Speicherplatz 4004 gespeichert werden.

lfd.Nr.	Adr.-Teil	Op.	Erläuterung	
4000			= a	
4001			= b	
4002			= c	
4003			= d	
4004			= X	(algebraische Summe)
4005	4000	215	⟨AC⟩ := 2a	(lesen von a und Verschiebg.)
4006	4001	210	⟨AC⟩ :=2a-b	(um eine Dualstelle n. links)
4007	4002	220	⟨AC⟩ :=2a-b-c	
4010	4003	210	⟨AC⟩ :=2a-b-c-d	
4011	4004	240	⟨4004⟩ := ⟨AC⟩ := 2a-b-c-d	
4012		020	unbedingter Stopp	

Anstelle der Operation 215 zum Lesen von 2a (BAZ = 215, Lesen mit Linksverschiebung um ein Bit ≙ Multiplikation mit 2 im Dualsystem) kann die Bildung von 2a auch durch wiederholte Addition von a erfolgen.

Beispiel 3: Multiplikation einer Zahl x mit 10

Programmbeginn: 2000

Der Wert x steht auf dem Speicherplatz 7700, das Er-

gebnis soll zum Speicherplatz 7701 transportiert werden. Es sollen die Möglichkeiten der Resultatbehandlung bei den Einzelbefehlen ausgenutzt werden.

lfd.Nr.	Adr.-Teil	Op.	Erläuterung
2000	7700	211	< AC > := x
2001	7700	214	< AC > := (x+x)2= 4x
2002	7700	214	< AC > := (4x+x)2= 10x
2003	7701	240	< 7701 > := 10x
2004		020	unbedingter Stopp

Bemerkung: Es ist zu beachten, daß die Resultatbehandlung nach Ausführung der Operation erfolgt und sich auf den < AC > bezieht.

Beispiel 4: Ausblenden von Adressenstellen

Programmbeginn: 4000

Die Adressenstellen a eines Maschinenwortes auf dem Speicherplatz 3001 sollen in die Adressenstellen a eines auf dem Speicherplatz 3000 stehenden Lesebefehls eingeschrieben werden.

< 3000 > = xxxx 211

< 3001 > = a_1 a_2 a_3 a_4 xxx

Auf den Speicherplätzen 3002 und 3003 stehen folgende Konstanten (Masken):

< 3002 > = 0000777

< 3003 > = 7777000

lfd.Nr.	Adr.-Teil	Op.	Erläuterung
4000	3002	211	< AC > := 0000777
4001	3000	250	< 3000 > := 0000211
4002	3003	211	< AC > := 7777000
4003	3001	200	< AC > := a_1 a_2 a_3 a_4 000
4004	3000	210	< AC > := a_1 a_2 a_3 a_4 211
4005	3000	240	< 3000 > := a_1 a_2 a_3 a_4 211
4006		020	unbedingter Stopp

Beispiel 5: Betragsbildung einer Summe

Programmbeginn: 3000

Zwei auf den Speicherplätzen 4000 und 4001 stehende Festkommazahlen a und b sollen summiert und der Betrag nach 4002 transportiert werden.

S = a+b

< 4000 > = a
< 4001 > = b
< 4002 > = S

lfd.Nr.	Adr.-Teil	Op.	Erläuterung
3000	4000	211	< AC > := a
3001	4001	210	< AC > := a+b
3002	3005	050	Sprung, wenn a+b positiv
3003	4002	240	< 4002 > := a+b (negativ)
3004	4002	221	< AC > := a+b
3005	4002	240	< 4002 > := a+b
3006		020	unbedingter Stopp

Beispiel 6: Addition mit Zahlenbereichsüberschreitung und Textausdruck

Programmbeginn: 4000

Zwei auf den Speicherplätzen 3500 und 3501 stehende Festkommazahlen a und b sollen summiert und das Ergebnis zum Speicherplatz 3502 transportiert werden. Falls bei dieser Addition der Zahlenbereich überschritten wird (Überlauf), soll über das Schreibwerk "Überlauf" ausgeschrieben werden.

< 3500 > := a
< 3501 > := b
< 3502 > := Ergebnis (wenn kein Überlauf erfolgt)

```
lfd.Nr.  Adr.-Teil  Op.  Erläuterung

4000     3500       211  < AC >  := a
4001     3501       240  < AC >  := a+b
4002     4005       070   SPRUNG, wenn durch a+b Überlauf
4003     3502       240  < 3502>:= a+b
4004                020   unbedingter Stopp
4005     4500       030   Anwahl Kanal 2
4006     0570       030   Umschaltung GB
4007     1440       030   Druck "U"
4010     1170       030   Umschaltung KB
4011     0450       030   Druck "e"
4012     0620       030   Druck "b"
4013     0450       030   Druck "e"
4014     1110       030   Druck "r"
4015     1030       030   Druck "l"
4016     0610       030   Druck "a"
4017     1440       030   Druck "u"
4020     0460       030   Druck "f"
4021                020   unbedingter Stopp
```

Beispiel 7: Ermittlung des Maximums dreier Festkommazahlen unter Anwendung der Unterprogrammtechnik

Programmbeginn Hauptprogramm: 3000

Programmbeginn Unterprogramm: 4000

Auf den Speicherplätzen 6000 bis 6002 stehen die Festkommazahlen a,b,c. Der größte Wert soll zum Speicherplatz 6003 transportiert werden. Das Bestimmen des größten Wertes soll von einem Unterprogramm vorgenommen werden. Laut Unterprogrammtechnik werden dem Unterprogramm die Parameter von der äußeren Adresse 1 des Unterprogramms an durch das Hauptprogramm übermittelt. Ist das durch das Unterprogramm ermittelte Ergebnis eine einzelne Information, so muß sie beim Rücksprung in das Hauptprogramm im AC stehen.

```
< 6000 >:= a
< 6001 >:= b
< 6002 >:= c
< 6003 >:= Maximum
```

lfd.Nr.	Adr.-Teil	Op.	Erläuterung
Hauptprogramm			
3000	6000	211	⟨AC⟩ := a
3001	4001	240	Transp. zur Adr. 1 des UP
3002	6001	211	⟨AC⟩ := b
3003	4002	240	Transp. zur Adr. 2 des UP
3004	6002	211	⟨AC⟩ := c
3005	4003	240	Transp. zur Adr. 3 des UP
3006	4000	025	Sprung mit Rücksprungadresse ins UP
			⟨AC⟩ := 3007 000
3007	6003	240	⟨6003⟩:= Maximum (Transport erfolgt nach Abarbeitung des UP)
3010		020	unbedingter Stopp
Unterprogramm			
4000	4004	000	Sprung über die Parameter
4001	Wert a		
4002	Wert b		
4003	Wert c		
4004	4014	240	Transp. von ⟨AC⟩:= 3007 000 an das Ende des UP 1 (Rücksprungsicherung)
4005	4001	211	⟨AC⟩:= a
4006	4002	330	Vergleich ⟨AC⟩/⟨4002⟩
4007	4011	150	Sprung bei ≥
4010	4002	211	⟨AC⟩:= b
4011	4003	330	Vergleich ⟨AC⟩/⟨4003⟩
4012	4014	150	Sprung bei ≥
4013	4003	211	⟨AC⟩:= c
4014	3007	000	Rücksprung ins HP

3. Die Programmierung des C 8205 mit Hilfe von Interpretiersystemen

3.1. Die Notwendigkeit der Anwendung von Interpretiersystemen zur rationellen Programmierung des Rechners

Bei der konstruktiven Gestaltung des Rechners wurde davon ausgegangen, bei einem Minimum an technischem Aufwand eine große Leistungsfähigkeit zu erreichen. Das wurde durch Mehrfachausnutzung von Baugruppen und Verzicht auf Verdrahtung von Multiplikation und Division erreicht. Die verhältnismäßig große Speicherkapazität des Rechners und die relativ hohe Rechengeschwindigkeit erlauben jedoch, Multiplikation, Division, Ein- und Ausgabe von Dezimalzahlen und die Gleitkommarechnung rationell durch Unterprogrammsysteme zu reali-

sieren. Für den Programmierer stellt die Anwendung der Unterprogramme keine besondere Schwierigkeit dar. Durch den Befehl "Sprung mit Rückkehrabsicht" beschränkt sich die Handhabung der Unterprogramme des "BIS" (siehe Pkt. 3.2.) beispielsweise auf ein bis drei Befehle. Man braucht dabei den Ablauf der Unterprogramme nicht zu kennen.

3.2. Übersicht über vorhandene Interpretiersysteme

Nach dem derzeitigen Stand sind folgende Interpretiersysteme in Anwendung:

1. BIS — Bausteininterpretiersystem
2. FIPS 1 — Festkommainterpretiersystem
3. GIPS — Gleitkommainterpretiersystem
4. FIPS-V — Festkommainterpretiersystem mit variabler Wortlänge

Außerdem besteht die Möglichkeit der Programmierung mit Hilfe von:

PROSA — Programmiersystem mit symbolischer Adressierung (Assembler) und PS-2.

3.3. Das Bausteininterpretiersystem (BIS)

Das Bausteininterpretiersystem "BIS" stellt ein offenes Unterprogrammsystem dar, das alle Routinen für Ein- und Ausgabe von Dezimalzahlen, für Multiplikation und Division von Festkomma- und Gleitkommazahlen sowie Addition und Subtraktion von Gleitkommazahlen und eine Reihe von Serviceprogrammen enthält. Alle Bausteine sind so aufgebaut, daß sie zeitoptimal arbeiten. Ihre Länge liegt zwischen 67_8 und 377_8 Speicherplätzen. Bei der für Kleinrechner relativ hohen Rechengeschwindigkeit ist der Zeitaufwand für die Abarbeitung der Bausteine z.B. gegenüber der Ein- bzw. Ausgabe gering. Das "BIS" ist so konzipiert, daß mit einigen Einschränkungen die einzelnen Bausteine nach Bedarf zusammengestellt und auf beliebigen Bereichen der Magnettrommel abgespeichert werden können. Dabei ist gewährlei-

stet, daß alle Bausteine trotz der frei wählbaren Abspeicherung unter festen Adressen erreichbar sind, wodurch dem Programmierer die Arbeit erleichtert wird. Er braucht die Abspeicheradressen nicht zu kennen. "BIS" wird vom VEB Kombinat Zentronik zur Verfügung gestellt. Es besteht aus einer Reihe von Bausteinen, die in vier Gruppen eingeteilt werden können:

Eingabebausteine
Rechenbausteine
Ausgabebausteine
Serviceprogramme

Dieses System gestattet, sowohl mit Festkommazahlen als auch mit Gleitkommazahlen zu arbeiten. Die praktische Programmierung erfolgt in der Maschinensprache. Es eignet sich grundsätzlich für die Bearbeitung wissenschaftlich-technischer, mathematischer und mit Einschränkungen auch für ökonomische Probleme.

3.3.1. Beschreibung der Bausteine

3.3.1.1. Das kleine Eingabeprogramm (KEP)

Das "KEP" befindet sich auf der Spur Ø der Trommel und belegt die Adressen $0000_8 \ldots 0037_8$. Dieses Programm ermöglicht die Eingabe leistungsstarker Programme als bitweise notierte und abgelochte Programme. Ohne dieses Programm wäre die Eingabe nur mit technischen Mitteln unter großem Aufwand möglich. Das "KEP" sollte durch Sperrung der Spur Ø grundsätzlich gegen Überschreiben geschützt sein. Es ist systemunabhängig und kann daher von allen Systemen gleichermaßen verwendet werden. Das "KEP" verarbeitet nur die Ziffern "Ø", "1", die Zeichen Punkt ".", Komma ",", Doppelpunkt ":" und Zwischenraum "ZR". Alle Wörter müssen demzufolge dual dargestellt sein. Mit Hilfe des KEP wird das auf dem Lochband dargestellte Minimalprogramm, Grundbaustein des "BIS", eingegeben.

3.3.1.2. Das Minimalprogramm

Da beim C 8205 nur die Eingabe eines Zeichens fest verdrahtet ist, sind zur Eingabe von Worten (Befehle, Zahlen oder Adressen) Leseprogramme erforderlich. Das Minimalprogramm enthält solche Routinen bzw. stellt den Anschluß zu anderen Bausteinen des "BIS" her. Mit Hilfe des Minimalprogramms werden Adressen und Programme eingegeben. Die Darstellung dieser Informationen erfolgt mit wenigen Ausnahmen im Oktalsystem. Bei der Eingabe unterscheiden wir zwischen der Eingabe von Adressen, Befehlen und Konstanten, die mit den einzugebenden Programmen in Verbindung stehen, und zu verarbeitenden Daten (im allgemeinen Dezimalzahlen). Das Minimalprogramm steuert dabei nur die Eingabe von Befehlen, Adressen und Konstanten. Die Eingabe von zu verarbeitenden Daten erfolgt über die Anwenderprogramme durch spezielle Eingabebausteine.
Zum Unterscheiden zwischen Adressen, Befehlen und Konstanten wurden besondere Eingabekennzeichen geschaffen. Diese Kennzeichen K_1 und K_2 gehen jeder Eingabe über das Minimalprogramm voran. Die externe Notierung eines Befehls sieht deshalb vollständig folgendermaßen aus:

K_1 K_2 a_1 a_2 a_3 a_4 BAZ.

K_1 ist das Unterscheidungsmerkmal für die verschiedenen möglichen Arten einer Information.

Folgende Ziffern und Zeichen sind als 1. Kennzeichen definiert:

die Ziffern: Ø, 1, 2, 3, 4, 5, 6, 7, 8, 9,
die Zeichen: B, D, +, Zwischenraum (ZR), Schrägstrich (/) und Doppelpunkt (:)

Das zweite Kennzeichen K_2 dient bei Befehlen der Adressenmodifikation 1. Art. Alle definierten Kennzeichen gelten gleichermaßen für die Eingabe im R 300- und Fernschreibcode.
Eingabekennzeichen werden nicht abgespeichert. Sie steuern lediglich die Eingabe.

$K_1 = \emptyset$ **Eingabe eines Maschinenwortes zur fortlaufenden Abspeicherung**

Nach der Eingabe eines zweiten Kennzeichens können maximal 11 Oktalziffern (z.B. Befehl) eingegeben werden. Die Eingabe wird durch das Zeichen Punkt "." abgeschlossen. Soll in der Reihenfolge der fortlaufenden Abspeicherung eine Speicherzelle frei gehalten werden (auf Null definiert werden), so genügt die Darstellung Ø (Null - Punkt).
Die Interpretation der so abgespeicherten Oktalzahl als Befehl oder beliebige andere oktale Konstante wird durch das Verarbeitungsprogramm bestimmt.

$K_1 = 1$ **Maschinenbefehl zur sofortigen Abarbeitung**

Nach Eingabe eines zweiten Kennzeichens erfolgt die Eingabe eines Maschinenbefehls (z.B. Sprungbefehl zum Start von Programmen).
Dieser Befehl wird zunächst in einer festen Speicherzelle des Systems gespeichert und nach Eingabe des Schlußzeichens "." sofort ausgeführt. Ist der Sofortbefehl kein Sprungbefehl, geht der Rechner nach Abarbeitung des Sofortbefehls in den Zustand "UU" (unbedingter Stopp).

Externe Darstellung: $1K_2\ s_1\ s_2\ s_3\ s_4$ BAL.

$K_1 = 2$ **Adresseneingabe**

Die Adresseneingabe ist notwendig, um z.B. die absolute Anfangsadresse eines Programms festzulegen, die für die Abspeicherung von Informationen verantwortlich ist.
Mit Hilfe des zweiten Kennzeichens K_2 unterscheidet das Minimalprogramm, um welche Art von Adresse es sich handelt. Dabei ergeben sich hier wie auch bei den beiden vorher erläuterten Kennzeichen folgende Bedeutungen für K_2:

$K_2 = \emptyset$ **Relativadresse**

Es wird keine Adressenmodifikation durchgeführt. Der Adreßteil des Befehls bleibt unverändert.

$K_2 = 1$ **Hauptleitadresse**

Sie legt die Anfangsadresse für die Abspeicherung von Informationen fest. Außerdem wird eine Adressenmodifikation mit dem Adreßteil des entsprechenden Maschinenwortes durchgeführt, wenn das 2. Kennzeichen = 1 ist. Die Hauptleitadresse (= 1. Leitadresse) wird zu den Oktalstellen $t_3 \ldots t_{10}$ des Maschinenwortes vor der Abspeicherung addiert. Ist $K_2 = 2 \ldots 9$ wird zu den Oktalstellen $t_3 \ldots t_{10}$ (Adreßteil des Befehls) die 2. ... 9. Leitadresse addiert.
Die Kennzeichen $K_1 = 3 \ldots 9$ dienen der Eingabesteuerung von Konstanten verschiedener Darstellungsarten. Hierbei folgt dem Kennzeichen K_1 kein zweites Kennzeichen, sondern unmittelbar die jeweilige Konstante (keine Adreßrechnung).

$K_1 = 3$ Eingabe einer Konstanten in Form einer Dezimalzahl (intern: Konvertierung in eine Zahl in Gleitkommaderstellung)

Externe Darstellung: 3 (-) a.

$K_1 = 4$ Eingabe einer Konstanten in Form einer Gleitkommazahl

Externe Darstellung: 4 (-) d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8.(-)e.

(Vgl. Pkt. 1.3.4. Zahl in Gleitkommadarstellung)

$K_1 = 5$ Eingabe einer Konstanten in Form einer Dezimalzahl (intern: Konvertierung in eine Festkommazahl, Komma hinten)

Externe Darstellung: 5 (-) a_9 a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0.
(Tab)

$K_1 = 6$ Eingabe einer Konstanten in Form einer Dezimalzahl (intern: Konvertierung in eine Zahl in Tetradendarstellung) Es können nur ganze positive Zahlen eingegeben werden.

Externe Darstellung: 6 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0.

$K_1 = 7$ Die Wirkung von $K_1 = 7$ entspricht der von $K_1 = 5$. Die Lesegeschwindigkeit ist hier höher.

Externe Darstellung: 7(-) s_7 s_6 s_5 s_4 s_3 s_2 s_1 s_0. (Tab)

Für $K_1 = 5, 6, 7$ gilt, daß ein eventuell vorhandenes Komma überlesen wird. Die Zahl wird in jedem Falle als ganze Zahl interpretiert und in der entsprechenden Form abgespeichert.

$K_1 = 8$ Eingabe von Dezimalzahlen in Satz (ohne Kennzeichen) (intern: Konvertierung in Gleitkommazahlen) Eingabe-Kennzeichen nur vor dem ersten Wort (siehe $K_1 = 3$)

$K_1 = 9$ Eingabe von Dezimalzahlen in Satz (ohne Kennzeichen) (intern: Konvertierung in Festkommazahlen, Komma hinten) Eingabe-Kennzeichen nur vor dem ersten Wort (siehe $K_1 = 5$)

Die einzugebenden Zahlen müssen gleichen Type sein. Sie sind durch Punkt "." (= Wortmarke) voneinander getrennt. Der Satz endet mit einer Satzmarke (WRHL) bzw. im Fernschreibcode mit Doppelpunkt ":".
Die Abspeicherung wird durch die Hauptleitadresse bestimmt, die dem Rechner vorher eingegeben werden muß. Sie endet mit einer Satzmarke. Der Rechner geht anschließend zur Eingabe eines neuen ersten Kennzeichens über.

$K_1 = B$ bzw. D Serviceprogramme

Über diese Kennzeichen werden Serviceprogramme erreicht (z.B. Bereichsausschreiben, Bereichseinstanzen, Protokollprogramm). Die Unterscheidung der einzelnen Serviceprogramme erfolgt durch nachfolgende Eingabe eines zweiten Kennzeichens K_2.

K_1	Bedeutung:	Eingabebild:
0	Maschinenwort zur fortlaufenden Speicherung	OK_2 xxxxxxxxxx.
1	Maschinenbefehl zur sofortigen Abarbeitung	$1K_2\ a_1a_2a_3a_4$ BAZ.
2	Adresseneingabe	$2K_2\ a_1a_2a_3a_4$.
3	Eingabe einer Dezimalzahl, Konvertierung in eine Gleitkommazahl	3(-) a.
4	Eingabe einer Zahl in Gleitkommadarstellung	4(-) d.(-) e.
5	Eingabe einer Dezimalzahl, intern: Konvertierung in eine Festkommazahl	$5(-)\ d_9d_8d_7d_6d_5d_4d_3d_2d_1d_o$.
6	Eingabe einer Dezimalzahl, Konvertierung in eine Zahl in Tetradendarstellung	$6\ d_7d_6d_5d_4d_3d_2d_1d_o$.
7	Eingabe einer Dezimalzahl analog $K_1 = 5$	$5(-)\ d_9d_8d_7d_6d_5d_4d_3d_2d_1d_o$.
8	Eingabe von Dezimalzahlen im Satz, intern: Konvertierung in Gleitkommadarstellung	9 a.a.a.a.a.a.a.a.SM
9	Eingabe von Zahlen ohne Kennzeichen im Satz, intern: Konvertierung in Festkommazahlen	

K_2	Bedeutung:
0	Relativadresse bei der Adresseneingabe, keine Adressenmodifikation bei der Eingabe.
1	1. Leitadresse = Hauptleitadresse, Adressenmodifikation bei d. Eingabe v. Maschinenbefehlen mit d. 1. Leitadresse
2...9	2. ... 9. Leitadresse, Adressenmodifikation bei der Eingabe von Maschinenbefehlen mit der 2. ... 9. Leitadresse

3.3.1.3. Eingabebausteine

Alle Bausteine des "BIS" sind Unterprogramme. Durch sie wird eine Reihe von nicht fest verdrahteten Funktionen wie z.B. die Eingabe von Zahlworten oder Informationseinheiten (Sätzen) durch sinnvolle Anwendung der Maschinenbefehle realisiert. Die Rückkehr zum Anwenderprogramm (Hauptprogramm) erfolgt automatisch. Die Anwendung der Bausteine beschränkt sich daher im allgemeinen für den Programmierer auf die Anwendung von ein bis drei Befehlen innerhalb des Hauptprogramms. Man kann die jeweiligen Unterprogrammsprünge mit Makrobefehlen, wie sie in größeren Datenverarbeitungsanlagen Anwendung finden, vergleichen.
Stellvertretend für eine Reihe von Eingabebausteinen des "BIS" soll hier der Baustein B 2.2 näher beschrieben werden. Dieser Baustein hat zwei Grundfunktionen: Er ermöglicht die Eingabe einer Dezimalzahl und die Eingabe von Dezimalzahlen in Sätze ohne Kennzeichen (Informationseinheiten). In der praktischen Anwendung sind das die häufigsten Anwendungsfälle für die direkte Eingabe von zu verarbeitenden Daten. Außerdem steuert das Minimalprogramm bei der Eingabe mit Kennzeichen $K_1 = 5$ bzw. $K_1 = 9$ über den Baustein B 2.2 die Eingabe von Dezimalzahlen (z.B. Eingabe dezimaler Konstanten innerhalb der Befehlsfolge eines Programms).

Eingabe eines Wortes ohne Kennzeichen

Diese Funktion des Bausteins B 2.2 kann durch Ansprung der festen Adresse 7573 erreicht werden.

Externe Darstellung: 00 7573 025.

< AC > = d d = Dezimalzahl

Die Eingabe endet mit dem Wortendekennzeichen (Wortmarke) "." oder Tab. (Lochung "Tab."). Die Größe der einzugebenden Zahl ist auf maximal 10 Stellen begrenzt, wobei die absolute Größe durch
$d = 2^{31} - 1 = 2\,147\,483\,647$ gegeben ist
(vgl. Pkt. 1.3.3. Festkommadarstellung).

Die Eingabe kann dabei gleichberechtigt sowohl über Lochband als auch über Tastatur bzw. Schreibwerktastatur erfolgen. Das Eingabegerät wird durch eine vor der Eingabe zu programmierende oder manuell vorzugebende Kanalanwahl bestimmt.

Eingabe im Satz

Bei der Eingabe im Satz wird durch Ansprung des Bausteins B 2.2 auf der festen Adresse 7565 die Eingabe einer Informationseinheit erreicht. Die einzelnen Worte des Satzes sind dabei durch Wortmarke (Punkt bzw. Tab.) voneinander getrennt. Der Satz endet mit einer Satzmarke (WRZL = Satzmarke), z.B. $d_1.d_2.d_3. \ldots d_n.SM$

Die gesamte Datenfolge kann mit einer Blockmarke (im R 300-Code das Zeichen ":", z.B. bei Lochbandende) abgeschlossen werden.
Die Zeichen Satzmarke und Blockmarke werden durch den Baustein abgefragt und bestimmen (vgl. Beispiel) die Art des Rücksprungs zum Hauptprogramm.
Da die Abspeicherung des einzugebenden Satzes auf der Trommel erfolgt, muß dem Unterprogramm die Anfangsadresse, von der an der Satz abgespeichert werden soll, bekannt sein. Es ist deshalb erforderlich, diese Adresse in Form eines Abspeicherbefehls in die Zelle 7745 zu transportieren.

$\langle 7745 \rangle := a_1\ a_2\ a_3\ a_4\ 240.$

In der Praxis wird dieser Abspeicherbefehl in Form einer Konstanten im Konstantenteil des jeweiligen Anwenderprogramms definiert. Vor Ansprung der Adresse 7565 (Eingabe im Satz) muß diese Konstante (Abspeicherbefehl) in die Zelle 7745 transportiert werden.

Beispiel:

```
0000 01 0007 000.
0001 01 0004 240.   Konstante: Abspeicherbefehl
0002 00.
0003 00.
0004 00.            3 freie Speicherplätze für die Abspeicherung
0005 00.            eines Satzes mit beispielsweise 3 Worten
0006 00.
0007 88 4500 030.   Anwahl von Kanal 2 z.B. Leser 2
0010 88 0001 211.   Lesen des Abspeicherbefehls
0011 88 7745 240.   Transport des Abspeicherbefehls nach 7745
0012 88 7565 025.   UP-Ansprung "Eingabe im Satz"
0013               Rückkehr aus UP bei Satzmarke (WRZL) n+1
0014               Rückkehr aus UP bei Blockmarke ( : ) n+2
```

Eingabebausteine des "ETS":

B 2.1 Dieser Baustein ermöglicht die Eingabe von Dezimalzahlen, die intern in Gleitkommazahlen konvertiert werden.

B 2.2 Eingabe von Dezimalzahlen, intern: Konvertierung in Festkommazahlen, Komma hinten.

a) Eingabe einer Zahl in Dezimalschreibweise mit $K_1 = 5$

b) Eingabe von Zahlen im Satz mit $K_1 = 9$

c) Eingabe einer Dezimalzahl ohne Kennzeichen (Ansprungadresse: 7573)

$\langle AC \rangle = d_9\, d_8\, d_7\, d_6\, d_5\, d_4\, d_3\, d_2\, d_1\, d_0$.

d) Eingabe von Dezimalzahlen ohne Kennzeichen im Satz (Ansprungadresse: 7565) Die Speicherung erfolgt ab vorzugebender Anfangsadresse auf der Trommel.

B 2.3 Eingabe von Dezimalzahlen, intern: Konvertierung in Tetraden bzw. in Festkommazahlen.

a) Eingabe einer maximal 8-stelligen Dezimalzahl, die intern in die tetradische Darstellung konvertiert wird, mit $K_1 = 6$

b) Eingabe einer Dezimalzahl ohne Kennzeichen, intern: Konvertierung in die tetradische Darstellung (Ansprungadresse: 7570)

c) Eingabe einer maximal 8-stelligen Dezimalzahl mit $K_1 = 7$, intern: Konvertierung in Festkommadarstellung

d) Eingabe einer maximal 8-stelligen Dezimalzahl ohne Kennzeichen (Ansprungadresse: 7571)

Die Varianten c) und d) unterscheiden sich von den Varianten a) und b) des Bausteins B 2.2 dadurch, daß eine schnellere Eingabe mit Hilfe des Bausteins B 2.3 möglich ist. Die Stellenzahl ist dabei allerdings auf maximal 8 Stellen begrenzt.

B 2.4 Eingabeunterprogramm für die Eingabe von Dezimalzahlen mit Kommainformation

B 2.5 Eingabeunterprogramm für die Eingabe von Externbefehlen

Durch das Kennzeichen $K_1 = +$ können Externbefehle eingegeben werden, die bei der Eingabe interpretiert und in entsprechende Maschinenbefehle umgewandelt werden.

(Ausführlichere Informationen vgl. Heft 4 Bausteininterpretiersystem der Reihe "Bedienung und Programmierung des digitalen Kleinrechners C 8205" des VEB Kombinat Zentronik.)

3.3.1.4. **Ausgabebausteine**

Mit Hilfe von Ausgabe-Halt-Befehlen ist nur die Ausgabe von je einer Ziffer bzw. eines Zeichens pro Befehl möglich. Um die Ausgabe von Ziffernfolgen (Daten, Dezimalzahlen) rationell zu ermöglichen, sind spezielle Ausgabeunterprogramme geschaffen worden.

B 4.1 Dieser Baustein ermöglicht die Ausgabe des Inhalts von Speicherplätzen in oktaler Darstellung über den Stanzer.

a) UP-Ausgabe eines Wortes in Form einer Oktalzahl (Ansprungadresse: 7563)
Die auszugebende Zahl muß vorher in das Register 1 (R1) mit der Adresse 7773 transportiert werden.

b) Bereichsausstanzen (Serviceprogramm)
Mit diesem Programm können Speicherbereiche über LB-Stanzer in einer maschinenlesbaren Form ausgegeben werden (z.B. Ausgabe von Programmen). Eine genaue Beschreibung folgt unter Pkt. 3.3.1.6. (Serviceprogramme).

B 4.2 Ausgabe einer Gleitkommazahl in Gleitkommaschreibweise (Ansprungadresse: 7562)

B 4.3 Ausgabe einer Gleitkommazahl in Dezimalschreibweise

a) Ansprungadresse 7560, Ausgabe ohne Rundung
b), " 7561, " mit "

B 4.4 Ausgabe einer Festkommazahl, Komma hinten, in Form einer Dezimalzahl mit Komma (Variante 1)

B 4.5 Ausgabe einer Festkommazahl, Komma hinten, in Form einer Dezimalzahl mit Komma (Variante 2)

a) Ausgabe einer Festkommazahl in Form einer Dezimalzahl ohne Rundung (Ansprungadresse: 7555) Variante 2

b) Analog a), jedoch mit Rundung (Ansprungadresse: 7556)

c) Ausgabe einer Zahl in tetradischer Darstellung als Dezimalzahl (Ansprungadresse: 7554)

Für alle Bausteine der Ausgabe gilt, daß die auszugebende Zahl in aufbereiteter Form vor Ansprung des entsprechenden Bausteins in das Register 1 mit der absoluten Adresse 7773 transportiert werden muß. Wortendekennzeichen z.B. bei Ausgabe von Daten über LB-Stanzer werden dabei nicht berücksichtigt. Sie müssen separat durch Ausgabe-Halt-Befehle programmiert werden (außer B 4.1 Bereichsausstanzen).

Ermittlung der Kommainformation bei der Ausgabe durch die Bausteine

Variante 1

Über zwei Parameter p_1 und p_2 sind dem Ausgabe-Unterprogramm mit dem Ansprungbefehl die Informationen über die Stellung eines geforderten Kommas mitzuteilen. Es werden dazu die bisher nicht belegten Bitpositionen des Maschinenbefehls belegt.

Externe Darstellung: 00 b_1 b_2 b_3 b_4 a_1 a_2 a_3 a_4 025.

Für die Parameter gilt: $0 \leq p_1, p_2 \leq 15$

Die maximal 10 auszugebenden Dezimalziffern eines Maschinenwortes werden dabei wie folgt durchnumeriert:

$$d_9\ d_8\ d_7\ d_6\ d_5\ d_4\ d_3\ d_2\ d_1\ d_0.$$

Der Parameter p_2 bestimmt, wieviel Ziffern ab d_9 ausgegeben werden und vor dem Komma erscheinen sollen.

Der Parameter p_1 bestimmt, wieviel Ziffern nach dem Komma ausgegeben werden sollen.
Weil beim Druck stets mit d_9 begonnen wird und bei kleinen Zahlenwerten die Vornullen durch ZR ersetzt werden, ist die Variante 1 nicht zeitoptimal.

Da bei der Eingabe von Befehlen alle Zahlworte als Oktalzahlen interpretiert werden, müssen die Parameter p_2 und p_1, die die Bitpositionen $z_4 \ldots z_7$ und $z_8 \ldots z_{11}$

in tetradischer Form belegen sollen, entsprechend in eine gleichwertige oktale Darstellung umgewandelt und so notiert und abgelocht werden.

Variante 2

Für Variante 2 muß der Baustein B 4.5 abgespeichert sein.
Hierbei müssen 3 Parameter angegeben werden:

a) p_3 = erste zu druckende Ziffer (Index der Dezimalstelle 0 ... 9)
Bitposition: $z_0 \ldots z_3$

b) p_2 = Anzahl der Stellen vor dem Komma
Bitposition: $z_4 \ldots z_7$

c) p_1 = Anzahl der Stellen nach dem Komma
Bitposition: $z_8 \ldots z_{11}$

Die sich ergebende 3-stellige Dezimalzahl muß auch hier in eine maschinenlesbare Oktalzahl (Triadendarstellung) umgewandelt werden.
Durch die Angabe des Index der ersten zu druckenden Ziffer (p_3) übergeht der Rechner bei der Ausgabe die vor dieser Ziffer stehenden Zahlenwerte, und es kann mit Variante 2 eine Zeitoptimalität bei der Ausgabe erreicht werden.

Beispiel:

Inhalt des Maschinenwortes: 3 2 1 7 6 5 4

Druckbild : 3 2 1 7 6,5 4

p_3, p_2, p_1 : $p_3 = 6$, $p_2 = 5$, $p_1 = 2$ 652_{10}

Kommainformation : 3122_8

Verfahrensweise:

0	L	L	0	0	L	0	L	0	0	L	0	tetradisch
0	L	L	0	0	L	0	L	0	0	L	0	triadisch
	3			1			2			2		

Entsprechender Einsprungbefehl: 00 3122 7555 025.

Geht man von der Annahme aus, daß die auszugebende Information rechtsbündig in dem Maschinenwort gespeichert ist und keine Stelle rechts abgeschnitten werden soll, so ergibt sich: $p_2 + p_1 = p_3 - 1$

Beim Druck werden keine Vornullen ausgegeben. Vornullen werden durch Leerschritte ersetzt. Ein eventuelles Minuszeichen wird nach der Zahl gedruckt. Bei positiven Zahlen erfolgt dafür ein Leerschritt.

- Existieren Ziffern vor der ersten zu druckenden Ziffer $\neq 0$, wird zg (zu groß) gedruckt.

Hinweis:

Die Bausteine sind nicht für die Ausgabe von Daten über Lochbandstanzer geeignet, wenn diese Daten wieder eingelesen werden sollen. Das Zeichen Leerschritt (ZR = Zwischenraum), das anstelle von Minus bei positiven Zahlen ausgegeben wird, führt zur Löschung des AC bei der Eingabe (Korrekturzeichen "Irrung Wort"). Für einen solchen Anwendungsfall müssen die Ausgabebausteine verändert werden.

3.3.1.5. Rechenbausteine

Beim C 8205 sind nur Addition und Subtraktion von Festkommazahlen in ihren vielfältigen Variationen sowie die Eingabe eines Zeichens fest verdrahtet.
Multiplikation und Division von Festkommazahlen sowie Addition und Subtraktion von Gleitkommazahlen müssen wie Ein- und Ausgabe über Unterprogramme realisiert werden.

Festkommamultiplikation und Festkommadivision

Die Festkommamultiplikation und -division wird durch die Bausteine

B 5.4 Festkommamultiplikation und
B 5.5 Festkommadivision realisiert.

Es ergeben sich dabei folgende Varianten:

a) $\langle R_1 \rangle := \langle AC \rangle := \langle R_1 \rangle \times \langle R_3 \rangle \times 10^{-q_1}$ m.Rückspeicherung i.R_1

b) $\langle AC \rangle := \langle R_1 \rangle \times \langle R_3 \rangle \times 10^{-q_1}$

c) $\langle R_1 \rangle := \langle AC \rangle := \langle R_1 \rangle : \langle R_3 \rangle \times 10^{q_1}$ m.Rückspeicherung i.R_1

d) $\langle AC \rangle := \langle R_1 \rangle : \langle R_3 \rangle \times 10^{q_1}$

Durch Angabe des Parameters q, der in den Bitpositionen $z_0 \ldots z_2$ stehen muß, wird bei der Multiplikation eine Multiplikation des Ergebnisses mit 10^{-q} und bei der Division eine Multiplikation des Ergebnisses mit 10^{q} ausgeführt. Der Parameter q wird in oktaler Form mit dem Ansprungbefehl eingegeben.

Für q gilt: $0 \leq p_1 \leq 7$

Bei der Multiplikation wird er nach der Formel
$q = f_1 + f_2 - p$ ermittelt.

(f_1 = Stellenzahl des 1. Faktors nach dem gedachten Komma

f_2 = " " 2. " " " " "

p = gewünschte Stellenzahl des Produkts nach dem gedachten Komma)

Bei der Division wird der Parameter q nach der Formel
$q = d_2 - d_1 + u$ ermittelt.

(d_2 = Stellenzahl des Divisors nach dem gedachten Komma

d_1 = " " Dividenden " " " "

u = gewünschte Stellenzahl des Quotienten nach dem gedachten Komma)

Externe Darstellung des Einsprungbefehls:

00 q000 7543 025.

a) Festkommamultiplikation mit Speicherung in R_1 Adresse: 7543

b) Festkommamultiplikation ohne Speicherung " 7542

c) Festkommadivision mit Speicherung in R_1 " 7541

d) Festkommadivision ohne Speicherung " 7540

Der Zahlenbereich ist durch die Bedingung $z \leq 2^{31} - 1$ begrenzt. Bei Zahlenbereichüberschreitung (z.B. bei Division durch Ø) geht der Automat in den Zustand "Fehler Überlauf", und die Kontrollampen FÜ und BS des Anzeigeteils leuchten auf. Mit Hilfe des Befehls "Überlaufsprung" wird ein Sprung zu der im Adreßteil dieses Befehls bezeichneten Adresse ausgeführt, wenn der vorhergehende Befehl zur Kapazitätsüberschreitung führt und damit z.B. in ein Fehlermaßnahmeprogramm gesprungen werden soll. Wie aus der Darstellung bereits ersichtlich, müssen die entsprechenden Ausgangswerte für Multiplikation oder Division vor Einsprung in das jeweilige Unterprogramm in den Registern R_1 (Adresse 7773) und R_3 (Adresse 7776) bereitgestellt werden.

Gleitkommaoperationen

B 5.1 Gleitkommaaddition und Gleitkommasubtraktion

a) Gleitkommaaddition mit Speicherung des Ergebnisses in R_1

b) " ohne Speicherung

c) Gleitkommasubtraktion mit Speicherung des Ergebnisses in R_1

d) " ohne Speicherung

a) Ansprungzelle 7553 $\langle R_1 \rangle := \langle AC \rangle := \langle R_1 \rangle + \langle R_2 \rangle$

b) " 7552 $\langle AC \rangle := \langle R_1 \rangle + \langle R_2 \rangle$

c) " 7551 $\langle R_1 \rangle := \langle AC \rangle := \langle R_1 \rangle - \langle R_2 \rangle$

d) " 7550 $\langle AC \rangle := \langle R_1 \rangle - \langle R_2 \rangle$

B 5.2 Gleitkommamultiplikation

B 5.3 Gleitkommadivision

a) Gleitkommamultiplikation mit Speicherung Ansprungadresse 7547

b) " ohne Speicherung Ansprungadresse 7546

c) Gleitkommadivision mit Speicherung Ansprungadresse 7545

d) " ohne Speicherung Ansprungadresse 7544

Die Belegung der Register und die Verknüpfung der Operanden erfolgt analog zur Festkommamultiplikation bzw. -division.

3.3.1.6. Serviceprogramme

Der C 8205 verfügt über eine Reihe von Hilfsprogrammen, die die rationelle Testung, Dokumentation und Wartung von Programmen erleichtern.

B 6.2 Bereichsausschreiben

Durch dieses Programm ist es dem Programmierer möglich, den Inhalt von Speicherbereichen oder den Inhalt einer Speicherzelle über das Schreibwerk ausgeben zu lassen. Diese Programme werden über die Tastenkombination D3, D4, D5 erreicht (vgl. Pkt. 5.9.5.1.).

B 6.3 Vergleichsprogramm

Das Vergleichsprogramm ermöglicht den Vergleich einer Information auf dem Speicher mit einer Information, die über ein externes Gerät eingegeben werden muß. Das Programm wird durch die Tastenkombination B9 erreicht. Das jeweilige Eingabegerät (Kanalanwahl) ist manuell über Tastatur anzuwählen. Die Leitadresse der zu vergleichenden Information muß dem Automaten vorher eingegeben werden.

Bei Nichtübereinstimmung entsteht folgender Ausdruck:

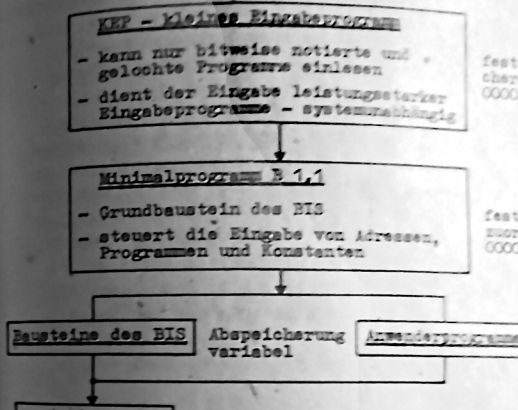
Adresse	Information extern	Information auf dem Speicher

Nach erfolgtem Ausdruck wird der Vergleich fortgesetzt. Wird zur Eingabe nicht Kanal 2 benutzt, so hält der Automat nach jeder Ausschrift an und muß durch H K_x R erneut gestartet werden (vgl. Pkt. 5.9.5.4.).

B 6.1 Protokollprogramm

Über das Protokollprogramm ist es möglich, ein Programm Befehl für Befehl abarbeiten zu lassen. Dabei wird nach Abarbeitung jedes Befehls der aktuelle AC-Inhalt in Form einer Dezimalzahl über das Schreibwerk ausgegeben. Die Zieladressen von Sprungbefehlen sowie UP-Ansprünge werden ebenfalls protokolliert. Das Protokoll wird über die Tastenkombination D7 erreicht (vgl. Pkt. 5.9.5.2.).

3.4. Darstellung der Zusammenhänge zwischen KEP, Minimalprogramm, Bausteinen und Anwenderprogrammen



3.4.1. Funktionsmechanismus

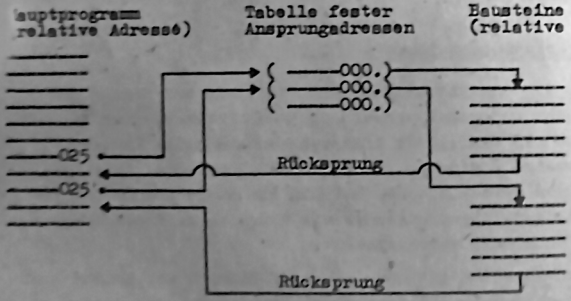
Das KEP (mit technischen Hilfsmitteln auf Spur Ø gebracht und gegen unbeabsichtigtes Überschreiben geschützt) ermöglicht eine systemunabhängige Eingabe bitweise notierter und gelochter Programme. Im allgemeinen werden mit dem KEP die Eingabeprogramme der jeweiligen Interpretiersysteme eingelesen. Diese Programme sind dual verschlüsselt.

Grundsätzlich ist aber auch die Eingabe beliebiger dual verschlüsselter Programme möglich. Wegen des großen Aufwandes scheidet diese Möglichkeit jedoch für die praktische Arbeit mit dem Rechner aus.

Das Minimalprogramm des BIS und die Eingabeprogramme der anderen Interpretiersysteme sind dual verschlüsselte Programme. Diese Programme brauchen im allgemeinen,

wenn man sich für die Arbeit in einem bestimmten Interpretiersystem entschieden hat, nur einmalig abgespeichert zu werden. Sie können dann gegen Überschreiben durch Sperrung der benutzten Speicherbereiche geschützt werden. Sowohl die Eingabe der anderen Bausteine, z.B. des BIS, wie die Eingabe der Anwenderprogramme wird dann ausschließlich durch das Minimalprogramm gesteuert.

Die Abspeicherung der Bausteine des BIS ist mit einigen Einschränkungen frei wählbar. Trotzdem sind diese Unterprogramme unter festen Einsprungadressen erreichbar. Die Bausteine sind so aufgebaut, daß bei der Eingabe des jeweiligen Bausteins die zugehörige wirkliche Einsprungadresse unabhängig von der Wahl der Abspeicheradresse für den Baustein in feste Ansprungzellen gebracht wird. Das erleichtert dem Programmierer die Arbeit. Er braucht die Abspeicherung der Bausteine nicht zu kennen. Die Speichereinteilung kann so durch den Operator unabhängig vom Programm optimal vorgenommen werden.



Der Einsprung in ein benötigtes Unterprogramm des BIS erfolgt durch einen Sprung mit Rückkehrabsicht zu einer für die gewünschte zu realisierende Funktion fest vorgegebenen Adresse (z.B. 7565 025. "Eingabe im Satz"). In dieser Zelle befindet sich ein Sprungbefehl zu der

durch die frei wählbare Abspeicherung bedingten variablen wirklichen Anfangsadresse des jeweiligen Bausteins. Der erste Befehl des entsprechenden Bausteins ist dann ein Befehl, der den AC-Inhalt sicherstellt. Er enthält die durch BAZ = 025. gebildete Rücksprungadresse zum Hauptprogramm, da durch nachfolgende Operationen innerhalb des UP die Rücksprungadresse verloren gehen würde (Überschreibung des AC-Inhalts). Auch die Betätigung der B- und D-Tasten zur Erreichung der Serviceprogramme führt zu festen Einsprungzellen, die beim Abspeichern der Bausteine ebenfalls automatisch mit den wahren Einsprung- bzw. Anfangsadressen der Bausteine gefüllt werden.

3.4.2. Anschlußbedingungen zu den Anwenderprogrammen

- Register, Anschluß- und Ansprungzellen, Kommunikation -

Alle Bausteine des BIS sind über feste Einsprungadressen durch die Anwenderprogramme erreichbar (vgl. Seite 151/152). Die Anwendung der Bausteine beschränkt sich dabei auf die Anwendung von ein bis drei Befehlen für den Programmierer. Unter Berücksichtigung der hohen Arbeitsgeschwindigkeit bei fest verdrahteten Operationen und der Tatsache, daß die zur Anwendung kommenden Bausteine zeitoptimal aufgebaut sind, ist der Zeitaufwand für die Abarbeitung der Bausteine relativ unbedeutend.

Der Informationsaustausch zwischen den Anwenderprogrammen und den Bausteinen erfolgt über Register. Diese Register sind vereinbarte Speicherzellen, die sich von den anderen Speicherzellen der Trommel nicht unterscheiden.

Register 1 Adresse 7773

Register 2 " 7776

Register 3 " 7774

Bei der Ausgabe einer Festkommazahl muß z.B. der Inhalt der auszugebenden Speicherzelle nach R 1 mit der Adresse 7773 vor Einsprung in das Druckprogramm transportiert werden. Analog dazu müssen bei Multiplikation und Division die Operanden in die vorgegebenen Register transportiert werden. Das Ergebnis wird dann im AC oder in einem der Register dem Hauptprogramm zur Verfügung gestellt.

Die Register können, wenn sie durch Bausteine nicht benutzt werden, durch den Programmierer beliebig verwendet werden.

3.5. Programmierbeispiele

Bei den folgenden Beispielen handelt es sich im Gegensatz zu den Programmierbeispielen von Pkt. 2.3. um relative ELS-Programme. Diese beginnen grundsätzlich mit der äußeren Adresse 0000 und weisen Eingabekennzeichen auf. Sie sind durch entsprechende Wahl der Haupteitadresse beliebig auf freien Speicherplätzen abspeicherbar.

Beispiel 1:
Eingabe, Maximumermittlung und Ausgabe des Maximums dreier Zahlen

Auf Lochstreifen sind drei Dezimalzahlen, jeweils durch eine Wortmarke (Punkt oder Tab.) voneinander getrennt und durch eine Satzmarke (WSM) abgeschlossen, abgelocht.
Die Dezimalzahlen sind ab 2. Leitadresse abzuspeichern (in interner Festkommadarstellung), und ihr Maximum ist zu bestimmen und auszudrucken.
Folgende Stellenzahl vor und nach dem Komma ist gegeben:

v = 3
n = 2

lfd.Nr.	KZ	Adr.-Teil	Op.	Erläuterungen
0000	00	4500	030.	Anwahl Leser 2 / UW = Kanal 2
0001	00	1170	030.	Umschaltung KB
0002	00	2720	030.	WSM1
0003	01	20	211.	⟨AC⟩ := Konstante für den Abspeicherbefehl
0004	00	7745	240.	⟨7745⟩:= " " "
0005	00	27564	025.	Eingabe im Satz mit Kommainformation
0006				
0007	02	0	211.	⟨AC⟩ := 1. Wert
0010	02	1	330.	Vergleich ⟨AC⟩/ 2. Wert
0011	01	12	150.	Sprung bei ⟨AC⟩ ≥
0012				
0013	02	1	211.	⟨AC⟩ := 2. Wert
0014	02	2	330.	Vergleich ⟨AC⟩/ 3. Wert
0015	02	15	150.	Sprung bei ⟨AC⟩ ≥
0016	00	2	211.	⟨AC⟩ := 3. Wert
0017	00	7773	240.	⟨7773⟩:= Maximum
0020	00	2062 7555	025.	Ausgabe FK-Zahl Variante 2 o. R.
	02	0	020.	US
		0	240.	Konstante zum Einstellen des Abspeicherbefehls

Beispiel 2:

Eingabe, Berechnung und Ausgabe eines Satzes

Auf einem Lochstreifen sind n positive Dezimalzahlen, jeweils durch eine Wortmarke (Punkt oder Tab.) voneinander getrennt, abgelocht.
Als Datenendezeichen wurde eine -1. verwendet, anschließend steht eine Satzmarke (WRZL).
Der Satz ist ab 3. Leitadresse abzuspeichern.
Sämtliche Werte sind mit einer im Programm bereitzustellenden Festkommakonstante K = 100,00 zu multiplizieren und anschließend über das Schreibwerk auszugeben.
Folgende Stellenzahlen vor und nach dem Komma sind gegeben:

	v	n
K	3	2
Eingabewerte	2	2
Ausgabewerte	4	2

Druckbild: W1
W2
W3
:
:
Wn

fd.Nr.	KZ	Adr.-Teil	Op.	Erläuterungen
0000	00	4500	030.	Anwahl Kanal 2
0001	00	1170	030.	Umschaltung KB
0002	01	27	211.	⟨AC⟩ := Konstante Lesebefehl
0003	01	7	240.	Transport auf einen freien Speicherplatz
0004	01	26	211.	⟨AC⟩ := Konstante für den Abspeicherbefehl
0005	00	7745	240.	⟨7745⟩ := " " "
0006	00	27564	025.	Eingabe im Satz mit Kommainformation
0007	0.			⟨AC⟩ := aktueller Wert
0010	01	23	040.	Sprung bei ⟨AC⟩ negativ
0011	00	7773	240.	⟨7773⟩ := aktueller Wert
0012	01	24	211.	⟨AC⟩ := Festkommakonstante
0013	00	7774	240.	⟨7774⟩ := "
0014	00	2000 7543	025.	Multiplikation mit Rückspeicherung
0015	00	2720	030.	WRZL
0016	00	2502 7555	025.	Ausgabe des berechneten Wertes
0017	01	7	211.	} Erhöhung des Lesebefehls für den aktuellen Wert
0020	01	25	210.	
0021	01	7	240.	
0022	01	7	000.	Zyklussprung
0023	00	0	020.	US
0024	5	100,00.		Festkommakonstante
0025	00	1000.		Adresseneins
0026	03	0	240.	Konstante zum Einstellen des Abspeicherbefehls
0027	03	0	211.	Konstante f.d.Lesen d. aktuellen Wertes (f.d.Wiederholbarkeit d.Prog

Beispiel 3:
Fakturierung

Von einer Anzahl Posten eines Artikels soll aus der Menge M und dem Einzelpreis EP der Preis P der einzelnen Posten errechnet und ausgedruckt werden. Sämtliche Preise sind zum Gesamtpreis P_{ges} zu summieren.
Die Mengenangaben M sind wortweise in Dezimaldarstellung auf einem Lochstreifen abgelocht, dessen Ende durch "0." gekennzeichnet ist.
Der Einzelpreis EP wird im Programm bereitgestellt.
Es soll die Überschrift "FAKTURIERUNG" gedruckt werden.

Druckbild:

FAKTURIERUNG

M1	P1
M2	P2
:	:
:	:
Mn	Pn
	P_{ges}

Kommainformationen:

		v	n
Eingabe	M	4	3
Ausgabe	M	4	3
Ausgabe	P	7	2
Ausgabe	P_{ges}	8	2

lfd.Nr.	KZ	Adr.-Teil	Op.	Erläuterungen
0000	00	4500	030.	Anwahl KZ
0001	00	2720	030.	WRZL
0002	00	0570	030.	Umschaltung GB
0003	00	0460	030.	F
0004	00	0610	030.	A
0005	00	1220	030.	K
0006	00	1630	030.	T
0007	00	1440	030.	U
0010	00	1110	030.	R
0011	00	0510	030.	I
0012	00	0450	030.	E
0013	00	1110	030.	R
0014	00	1440	030.	U
0015	00	1050	030.	N
0016	00	0670	030.	G
0017	00	1170	030.	Umschaltung KB
0020	01	45	241.	⟨Pges⟩ := 0
0021	00	2720	030.	WRZL
0022	00	3 7572	025.	Eingabe im Wort - Menge
0023	00	7773	340.	⟨7773⟩ := M Vergleich ⟨AC⟩/0
0024	01	38	170.	Sprung bei ⟨AC⟩ = 0
0025	00	3103 7555	025.	Ausgabe M
0026	00	1550	030.	Tab.
0027	01	44	211.	⟨AC⟩ := EP
0030	00	7774	240.	⟨7774⟩ := EP
0031	00	3000 7543	025.	Multiplikation M · EP
0032	01	45	210.	Aufsummierung von
0033	01	45	240.	Pges
0034	00	4162 7555	025.	Ausgabe Pges
0035	01	21	000.	Zyklussprung
0036	00	1720	030.	ZL
0037	00	1550	030.	Tab.
0040	01	45	211.	⟨AC⟩ := Pges
0041	00	7773	240.	⟨7773⟩ := Pges
0042	00	4602 7555	025.	Ausgabe P_{ges}
0043	00	0	020.	US
0044	5	10,50.		Festkommakonstante EP
0045	0.			Speicherplatz für P_{ges}

a.

a.1. Hinweise für die Anwendung der Assemblersprache PROSA

Zur Erleichterung der Programmierarbeit werden für die Datenverarbeitungssysteme symbolische Programmiersprachen entwickelt. Entsprechen diese in ihrer logischen Struktur der Maschinensprache, werden sie als Assemblersprache bezeichnet.
PROSA ist als Assemblersprache für den C 8205 zu werten, obwohl für die praktische Programmierung die Benutzung von Maschinenbefehlen neben den möglichen PROSA-Instruktionen (vgl. Befehlsliste) unumgänglich ist. Bei der Benutzung der Assemblersprache PROSA sind folgende Vorteile von Bedeutung:

1. Es besteht die Möglichkeit, anstelle der numerischen Adressen sinnfällige symbolische Adressen zu verwenden.

2. Die Korrektur bereits geschriebener Programme oder Programmteile ist ohne deren Änderung möglich.

3. Im Operationsteil der PROSA-Instruktionen finden mnemotechnische Abkürzungen* Anwendung.

4. Der Einsatz von Makroinstruktionen verringert den Umfang der symbolischen Programme.

Die durch die Verwendung von PROSA-Instruktionen entstandenen symbolischen Programme müssen für ihre Abarbeitung über den C 8205 in die für den Rechner verständliche Maschinensprache übersetzt werden. Dies wird von einem speziellen Übersetzungsprogramm für maschinenorientierte symbolische Programme, dem Assembler, vorgenommen.

Ein in symbolischer Form geschriebenes Programm heißt Ursprungsprogramm (Quellenprogramm) und wird durch die Übersetzung zu einem relativen Maschinenprogramm (Objektprogramm).

---

*Mnemotechnik oder Mnemonik: die Kunst, das Gedächtnis durch Merkworte zu unterstützen.

Weiterhin werden bei der Übersetzung alle Regelverletzungen gegen die Konstruktion der Assemblersprache PROSA (syntaktische Fehler) über die Schreibmaschine ausgegeben.

Bei dem Objektprogramm handelt es sich um ein wahlfrei abzuspeicherndes und im BIS abzuarbeitendes Maschinenprogramm. Da es sich bei dem Programmiersystem PROSA nicht um ein Interpretiersystem handelt, muß es bei der Abarbeitung des Objektprogramms nicht abgespeichert sein.

4.2. Format der Assemblersprache PROSA

Bei vorliegendem Programmablaufplan wird das Programm im Programmiersystem PROSA unter Beachtung der Regeln dieser Sprache geschrieben. Dazu stehen für die Programmierung vier verschiedene Sprachelemente zur Verfügung:

- Befehle
- Bereichsdefinitionen
- Konstantendefinitionen
- Zeichenketten

Sie bilden den Instruktionsvorrat des Programmiersystems PROSA und haben folgende allgemeine Struktur:

Name	T	Operation: Parameter	T	Operand	T	Kennzeichen	W
	A		A		A		R
	B		B		B		Z L

Das Zeichen Zwischenraum (ZR) ist nur in der Zeichenkette zugelassen.
Die Ziffer "Ø" wird vom Buchstaben "O" durch hinzufügen eines Querstriches unterschieden.

4.2.1. Befehle

Alle vier Teile dieser Instruktion können belegt werden. Dabei haben sie im einzelnen folgende grundlegende Bedeutung:

- Der Name dient zur Kennzeichnung eines Befehls.
- Der Operationsteil enthält den jeweiligen Operationscode.
- Im Operandenteil wird die Befehlsadresse festgelegt.
- Der Kennzeichenteil ist für das zweite Kennzeichen des Befehls vorgesehen.

4.2.1.1. Der Name

Es ist möglich, in jeder Instruktion einen Namen anzugeben. Wird ein solcher Name als Adresse im Operandenteil eines anderen Befehls verwendet, kann auf die durch den Namen gekennzeichnete Instruktion Bezug genommen werden. Die echten Speicheradressen brauchen also nicht bekannt zu sein, wenn man sich der symbolischen Adressierung bedient.
Damit diese sehr praktische Adressierungsart eindeutig bleibt, darf der gleiche Name innerhalb eines Programms im Namenteil nur einmal vorkommen. Die Übersichtlichkeit eines Programms wird verbessert, wenn die verwendeten Namen auf die Bedeutung der jeweiligen Instruktion hinweisen. So wäre z.B. bei einer Instruktion, die am Anfang (Einsprungstelle) eines Programmzyklus steht, der Name ZYKL sinnvoll. Der Assembler legt für alle im Programm vorkommenden Namen ein Adreßbuch an. Dort wird jedem Namen eine echte Speicheradresse zugeordnet. Im Verlauf der Übersetzung werden die symbolischen Adressen (Namen) durch die numerischen Adressen ersetzt.
Das Adreßbuch wird im Anschluß an den Assembler abgespeichert und kann, wenn der Assembler mit der kleinstmöglichen Ladeadresse geladen wurde (ab 500 oktal), maximal 1114 (dezimal) Namen enthalten.
Bei der Bildung eines Namens müssen folgende Regeln beachtet werden:

1. Ein Name besteht aus maximal 5 Zeichen.
2. Bei der Namensbildung dürfen nur die Buchstaben A...Z und die Ziffern 1...9 verwendet werden.

3. Das erste Zeichen muß ein Buchstabe sein.
4. Die Zeichenkombinationen AC, R1, R2, R3 sind als Name verboten.

Beispiele:

Name	Bemerkungen
Start	richtig
PS 1	richtig
3 BMG	falsch (erstes Zeichen kein Buchstabe)
KON, 3	falsch (enthält neben Buchstaben und Ziffern auch ein verbotenes Sonderzeichen)
ZYKLUS	richtig (das 6. Zeichen wird vom Assembler überlesen und nicht berücksichtigt)

4.2.1.2. Die Operation

Der Operationsteil eines Befehls beinhaltet den Operationscode und eventuell geforderte Parameter (vgl. dazu die Punkte 3.3.1.3. bis 3.3.1.5.). Als Operationscode sind alle Maschinenbefehle aus dem BIS sowie die in der PROSA-Befehlsliste (vgl. Pkt. 4.6.) enthaltenen symbolischen Codes zugelassen.
Die symbolischen Befehle aus der PROSA-Befehlsliste werden vom Assembler entweder in die entsprechenden Unterprogrammsprünge oder Ausgabe-Halt-Befehle des BIS übersetzt bzw. sie realisieren nach ihrer Übersetzung die Festkommaeins-Subtraktion und -Addition oder die Adresseneins-Addition und -Subtraktion.
Für die Bildung des Operationsteils gelten folgende Regeln:

1. Die Parameter müssen vom Operationscode durch das Zeichen (:) getrennt und jeweils durch das Zeichen (.) abgeschlossen werden.
2. Die geforderten Parameter müssen auch angegeben werden, wenn sie gleich Null sind.

Beispiele:

Operation	Operand	Bemerkung
240	7745	richtig
D		falsch (D = Division von Festkommazahlen, verlangt eine Parameterangabe)
M:0.		richtig (M = Multiplikation von Festkommazahlen, der geforderte Parameter wurde, obwohl er Null ist, angegeben)
MG		richtig (MG = Multiplikation von Gleitkommazahlen, Gleitkommaoperationen verlangen keine Parameterangabe)
M:2	ADR	falsch (Parameter ist nicht durch das Zeichen "." abgeschlossen)
EF:1.1.		falsch (EF = Eingabe von Festkommazahlen, verlangt nur eine Parameterangabe)

4.2.1.3. Der Operand

Im Operandenteil wird die Befehlsadresse festgelegt. Für die Befehlsadresse sind folgende Angaben erlaubt:

1. Symbolische Adresse
2. Symbolische Adresse ± Oktalzahl
3. Oktalzahl
4. ± Oktalzahl
5. leer
6. R1
7. R2
8. R3
9. AC

Für die innerhalb eines Programms auftretenden symbolischen Adressen (die nach den gleichen Regeln wie ein Name zu bilden sind) werden vom Assembler aus dem Adreßbuch die echten Maschinenadressen herausgesucht und eingesetzt.
Wird eine symbolische Adresse verwendet, die nicht im gesamten Programm als Name definiert wurde, so

findet der Assembler keine Maschinenadresse, und es erfolgt eine Fehleraussschrift über die Schreibmaschine. Steht im Operandenteil zusätzlich zur symbolischen Adresse eine vorzeichenbehaftete Oktalzahl, wird mit der aufgefundenen Maschinenadresse eine dem Vorzeichen entsprechende Adreßrechnung durchgeführt. Oktalzahlen werden vom Assembler nicht verändert. Sie ergeben in Abhängigkeit vom Inhalt der Kennzeichenspalte im Objektprogramm absolute oder relative Maschinenadressen. In der Assemblersprache PROSA hat ein Vorzeichen einer alleinstehenden Oktalzahl im Operandenteil die Bedeutung einer Sternadresse. Der Assembler setzt zunächst die Maschinenadresse des solcherart gekennzeichneten Befehls als Adresse ein. Die nachfolgende Oktalzahl ermöglicht im Anschluß daran, daß auf eine Instruktion relativ zu der Instruktion mit der Sternadresse Bezug genommen werden kann. Dabei sind die verschiedenen Übersetzungsverhältnisse der PROSA-Instruktion (vgl. dazu Pkt. 4.2.1.5.) sowie das Vorzeichen der Oktalzahl zu beachten.

Beispiele zur Sternadresse:

Operation	Operand	Bemerkung
.	.	.
.	.	.
.	.	.
EF		Eingabe einer Festkommazahl
330	KON 1	Vergleich mit einer Konstanten
140	-2	Ist der Eingabewert kleiner, wird zur Eingabe zurückgesprungen.
.	.	.
.	.	.
.	.	.
EF		Eingabe einer Festkommazahl
330	KON 2	Vergleich mit einer Konstanten
140	+4	Ist der Eingabewert kleiner, wird die Erhöhung des Zählers übersprungen.
ADD1	ZELER	Erhöhung des Zählers (die Instruktion wird im Verhältnis 1:3 übersetzt)
.	.	.
.	.	.
.	.	.

Wird der Operandenteil eines Befehls leer gelassen, so bezieht sich der Befehl in Abhängigkeit vom Operandencode auf den AC oder auf die drei Register.
Die Angaben AC, R1, R2, R3 haben im einzelnen folgenden Bezug:

AC entspricht dem Akkumulator
R1 entspricht dem Rechenregister 1 = Speicherplatz 7773
R2 entspricht dem Rechenregister 2 = Speicherplatz 7776
R3 entspricht dem Rechenregister 3 = Speicherplatz 7774

4.2.1.4. Das Kennzeichen

Der Assembler realisiert im wesentlichen selbständig den Inhalt der Kennzeichenspalte, der dem 2. Kennzeichen der Befehle im BIS entspricht. Er verwendet dabei je nach Inhalt der Instruktion das Kennzeichen "0" für die absolute Adressierung oder das Kennzeichen "1" für die relative Adressierung (Adressenänderung 1. Art mit der Hauptleitadresse). Dadurch sind alle entstehenden Objektprogramme relative BIS-Programme, d.h., sie beginnen bei der Adresse 0000 und können in Abhängigkeit von der vorgegebenen Hauptleitadresse wahlweise im Hauptspeicher abgespeichert werden.
Für die Programmierung stehen die Kennzeichen 0 und 2...9 zur Verfügung. Bei ihrer Verwendung gelten folgende Regeln:

1. Die Kennzeichenspalte darf nur belegt werden, wenn im Operandenteil eine Oktalzahl ohne Vorzeichen steht.

2. Die Verwendung des Kennzeichens "1" ist für die Programmierung verboten.

Werden neben dem Kennzeichen "0" noch andere Kennzeichen angegeben, so müssen vor der Abspeicherung des Objektprogramms die den Kennzeichen 2...9 entsprechenden Leitadressen eingegeben werden.

4.2.1.5. Das Übersetzungsverhältnis

Die symbolischen Befehle der Assemblersprache PHOCA

werden in Abhängigkeit vom Operationscode in der Adress-
senangabe im Operandenteil im Verhältnis 1:1 bis 1:3
übersetzt. Bei den Übersetzungsverhältnissen 1:2 und
1:3 kann von Makroinstruktionen gesprochen werden.
Entsprechend der Operationsart ergeben sich folgende
Übersetzungsvarianten:

Variante 1: Unterprogrammsprünge für Rechen- und Ausgabeoperationen

Operation	Operand	Übersetzung	Übersetzungsverhältnis (Üv)
Op	leer	75xx 025	1:1
Op	AC	Rx 240	
		75xx 025	1:2
Op	Adr	Adr 211	
		Rx 240	
		75xx 025	1:3

Variante 2: Festkommaeins- und Adresseneins-Addition und -Subtraktion

Operation	Operand	Übersetzung	Übersetzungsverhältnis
Op	leer	5 210	1:1
Op	AC	5 210	1:1
		Adr 211	
Op	Adr	5 210	1:3
		Adr 240	
		(bei 1°-Addition)	

Variante 3: Eingabeoperation

Operation	Operand	Übersetzung	Übersetzungsverhältnis
Op	leer	75xx 025	1:1
Op	AC	75xx 025	1:1
		75xx 025	
Op	Adr	Adr 240	1:2

Variante 41 Anzahl- und Schreiboperationen

Operation	Operand	Übersetzung	Übersetzungsverhältnis
Op	leer	xxxx 030	1:1

Zeichenerklärung:

75xx = Adresse des jeweiligen Unterprogrammsprungs
Rx = Register 1, 2 oder 3
Adr = Symbolische Adresse, symbolische Adresse ± Oktalzahl, Oktalzahl

4.2.2. Bereichsdefinitionen

Für Bereichsdefinitionen stehen zwei Möglichkeiten zur Verfügung:

1. B Bereich freihalten
2. BN Bereich auf Null löschen

Im Teil "Name" kann eine symbolische Adresse angegeben werden, mit der der Bereich innerhalb des Programms angesprochen werden kann.
Durch die Angabe B im Operationsteil wird Speicherplatz reserviert, ohne dessen Inhalt zu verändern. Durch die Angabe BN wird der Speicherplatz zusätzlich mit Nullen belegt.
Im Operandenteil wird durch eine Oktalzahl die Anzahl der zu reservierenden Speicherplätze festgelegt.
Der Kennzeichenteil bleibt frei, so daß unmittelbar nach der Operandenangabe "WORT" gegeben wird.
Bei der Angabe B wird durch die Summe aus dem Befehlszählerstand bei der Übersetzung und der Oktalzahl des Operanden teils vom Assemblerprogramm mit Hilfe des Kennzeichens 20 (Relativadresse des BIS) die folgende Adresse bestimmt.
Bei der Angabe BN wird der Speicherplatz reserviert durch die Eingabe einer von der Oktalzahl bestimmten Anzahl von Zeichenkombinationen "0".

4.2.3. Konstantendefinitionen

- Folgende Konstanten können in dem Programmiersystem PROSA definiert werden:

Operationsteil	Bemerkungen	
KF	Konstante Festkomma	(= Kennzeichen 5)
KTRI	Konstante Triaden	(= Kennzeichen 6)
KG	Konstante Gleitkomma	(= Kennzeichen 3)
KGL	Konstante Gleitkomma halblog.	(= Kennzeichen 4)
KTET	Konstante Tetraden	(= Kennzeichen 6)

Bei der Übersetzung werden anstelle der mnemotechnischen Abkürzungen im Operationsteil die für die Eingabesteuerung im Minimalprogramm notwendigen Kennzeichen eingesetzt.
Die Zahlenbereichsbegrenzung und die anzugebenden Vorzeichen entsprechen denen des BIS.

- Für die Definition von Konstanten gelten folgende Regeln:

1. Die Konstantenbezeichnung wird im Operationsteil der Instruktion angegeben.
2. Die Konstante steht im Operandenteil.
3. Der Namenteil wird in der Regel belegt sein, der Kennzeichenteil muß frei bleiben (vgl. Pkt. 4.2.2.)
4. Die Konstanten sind nicht durch einen Punkt abzuschließen. Die Begrenzung erfolgt durch das unmittelbar nach dem Operandenteil folgende "WRZL".
5. Für die Konstante "KGL" gilt, daß zwischen Mantisse und Exponent ein Punkt zu setzen ist.

Beispiele:

Operationsteil	Operandenteil	Bemerkungen	
KP	100	richtig	Die Übersetzung erfolgt in 5100.
KTKX	1000	richtig	Die Übersetzung erfolgt in 001000. (Adressenzeiger)
KG	10	richtig	Die Übersetzung erfolgt in 310.
KGL	3,-1	richtig	Die Übersetzung erfolgt in 43.-1. (Angabe von Mantisse und Exponent = 0,03)
KTST	298	richtig	Die Übersetzung erfolgt in 6298.
KP	2,5.	falsch	Die Übersetzung erfolgt in 52,5.. (Regel 4 wurde nicht beachtet)
KGL	-2 -1	falsch	Die Übersetzung erfolgt in 4 -2 -1. (Regel 5 wurde nicht beachtet)

4.2.4. Zeichenketten

Im BIS wird für die alphanumerische Ausgabe pro Zeichen ein Ausgabe-Halt-Befehl verwendet. Dafür muß aus einer Tabelle unter größerem Zeitaufwand die für das jeweilige Zeichen benötigte Ausgabeadresse herausgesucht werden.

Im Programmiersystem PROSA ist diesbezüglich eine wesentliche Programmiererleichterung vorhanden. Für maximal 63 Zeichen besteht die Möglichkeit, eine Zeichenkette zu programmieren. Der Assembler wandelt bei der Übersetzung jedes Zeichen in den entsprechenden Ausgabe-Halt-Befehl um, so daß hier Routinearbeit entfällt.

Für die Bildung der Zeichenkette gelten folgende Regeln:

1. Eine Zeichenkette ist mit einem Komma zu beginnen und zu beenden.

2. Bis auf folgende Ausnahmen können alle Zeichen der Schreibmaschinentastatur verwendet werden:

   a) "," als Zeichen für Beginn und Ende der Zeichenkette,

   b) "ß" wird vom Assembler wie "," gelesen, da es die gleiche Ein- und Ausgabeadresse besitzt (vgl. Schreibmaschinentastatur),

   c) ";" hat die gleiche Ein- und Ausgabeadresse wie "'", wobei diesem Zeichen eine Sonderfunktion zukommt (vgl. Pkt. 4.3.1.).

3. Bei einer Instruktion, die eine Zeichenkette enthält, kann der Namenteil belegt sein. Anschließend kommt nur noch die Zeichenkette mit maximal 63 Zeichen.

Beispiele:

Name	Operation	Operand
Titel	, Materialrechnung,	
AUS01	, FLÄCHENFÜLLUNG = ...,	
	af: 2.2.	ERGEB
AUS02	, ß,	

4.3. Herstellung der Datenträger

Die im Programmiersystem PROSA erstellten Programme müssen für eine zeitoptimale Eingabe auf Lochstreifen abgelocht werden, u. zw. im 8-Kanal-Code und nach gewissen Regeln.

4.3.1. Quellenprogrammstreifen

Für jede Instruktion eines Quellenprogramms sind prinzipiell die vier Teile Name, Operation, Operand,

Kennzeichen vorgesehen, die durch Tabulatorsprünge voneinander getrennt werden. Das Zeichen "WEIL" schließt die Instruktion ab. Um für die Schreibmaschinenausschrift, die neben dem Lochstreifen entsteht, ein übersichtliches Schreibbild zu erhalten, wird für das Setzen des Tabulatorstopp folgende Einteilung empfohlen:

Linker Randsteller +8, +16, +13,

Ist bei einer Instruktion der Namensteil nicht belegt, so wird dieser durch Anschlagen der Tabulatortaste übersprungen. Die Lochung "WEIL" kann bereits nach dem Operandenteil erfolgen, wenn der Kennzeichenteil frei bleibt. Ist das gesamte Quellenprogramm nach diesen Regeln abgelocht, wird das Zeichen ":" als Endzeichen gesetzt. Besteht ein Quellenprogramm aus mehreren Teillochbändern, so muß jedes Lochband, mit Ausnahme des letzten, mit dem Zeichen "," abgeschlossen werden. Das letzte Teillochband ist wiederum mit dem Zeichen ":" zu beenden. Für den Ablochvorgang ist grundsätzlich die Schreibmaschinenstellung "Kleinbuchstaben" zu wählen. Bei Ablochfehlern kann die betreffende Instruktion durch das Zeichen "Apostroph" korrigiert werden. Dies hat allerdings vor dem Zeichen "WEIL" zu erfolgen (bei Zeichenketten vor dem Zeichen "Komma"). Die so negierte Instruktion muß erneut abgelocht werden.

4.3.2. Datenstreifen

Beim Aufbau eines Datenstreifens brauchen Regeln des Programmiersystems PROSA nicht berücksichtigt zu werden. Der Datenstreifen ist lediglich auf die Eingabeoperationen des übersetzten ESS-Programms (Objektprogramm) abzustimmen. Das heißt, es ist zu beachten, ob eine Eingabe im Wort oder im Satz vorgenommen werden soll und dabei die Eingabe mittels Eingabekennzeichen oder UP-Ansprung verwendet wird.

4.4. Korrekturmöglichkeiten bei fehlerhaften Programmen

Wenn bei einem abgelochten Quellenprogramm nach der Übersetzung festgestellt wird, daß es noch Fehler enthält (vgl. Pkt. 4.5.4.), dann kann es mit Hilfe eines Korrekturprogramms ausgebessert werden. Das Korrekturprogramm gibt die Möglichkeit, ohne das gesamte Quellenprogramm umschreiben und neu ablochen zu müssen, Teile desselben wegzulassen, neue Befehle einzufügen und fehlerhafte Befehle durch richtige zu ersetzen. Das stellt gegenüber den numerisch adressierten und deshalb nur sehr umständlich zu korrigierenden BIS-Programmen einen wesentlichen Vorteil dar. Bei der Korrektur wird der Quellenprogrammstreifen über den Leser 1 und der jeweilige Korrekturstreifen über den Leser 2 gelesen. Der Stanzer gibt das korrigierte Quellenprogramm aus.
Das Korrekturprogramm ist relativ programmiert und belegt oktal 540 Speicherplätze. Bei seiner Abspeicherung muß beachtet werden, daß das Minimalprogramm geladen, der Bahnsperrenschalter 400 geöffnet und der Anfang einer geraden Spur gewählt wurde. Der Programmstart erfolgt mit "H G R B5". Das Korrekturprogramm verwendet die Zellen 7577 bis 7777 als Arbeitsspeicher. Ist vom Quellenprogramm das Schlußzeichen ";" erreicht, geht der Rechner in US. Durch Betätigen der Starttaste kann ein erneuter Korrekturlauf erfolgen.
Besteht das Quellenprogramm aus mehreren, durch das Zeichen "." abgeschlossenen Te llochbändern, so geht der Rechner auch bei Erkennen des Zeichens "." in den US. Durch Einlegen des nächsten Teillochbandes in Leser 1 und durch Betätigen der Tastenkombination "H M R" wird der Korrekturlauf fortgesetzt (bei Hauptleitadresse 4254).

4.4.1. Streichen falscher Instruktionen

Durch Ablochen des Zeichens "s" wird dem Korrekturprogramm mitgeteilt, daß die Korrekturmöglichkeit "Streichen" genutzt werden soll. Die Angabe eines Namens bez.

eines Namens + Oktalzahl (durch "WRZL" abzuschließen) nach dem Zeichen "s" zeigt an, bis zu welcher Instruktion das Quellenprogramm unverändert zu duplizieren ist. Anschließend wird durch das Ablochen einer Oktalzahl bzw. eines Namens (+ Oktalzahl), die jeweils das Endzeichen "WRZL" erhalten, festgelegt, wieviel Instruktionen zu streichen sind bzw. welche die letzte zu streichende Instruktion ist (diese Instruktionen werden vom Leser 1 überlesen). Anschließend wird weiter dupliziert.
Die Ablochbilder beider Möglichkeiten sehen folgendermaßen aus:

1. S Name + Oktalzahl W R Z L

Oktalzahl W R Z L

2. S Name + Oktalzahl W R Z L

Name + Oktalzahl W R Z L

Dabei ist zu beachten, daß anstelle von Name + Oktalzahl jeweils auch nur ein Name stehen kann.

4.4.2. Einfügen fehlender Instruktionen

Das Zeichen "e" bedeutet, daß die Korrekturmöglichkeit "Einfügen" zur Anwendung kommen soll. Die Angabe eines Namens bzw. eines Namens + Oktalzahl (durch "WRZL" abzuschließen) hat die gleiche Bedeutung wie beim Streichen.
Danach werden die in der üblichen Weise abgelochten Instruktionen, die insgesamt durch das Zeichen ":" abgeschlossen werden, in das neue Quellenprogramm ein-

gefügt (gestanzt).

4.4.3. Korrigieren fehlerhafter Instruktionen

Für die Korrekturmöglichkeit "Korrigieren" wird das Zeichen "k" abgelocht. Die nach dem Namen bzw. dem Namen + Oktalzahl (Bedeutung vgl. Streichen und Einfügen) kommenden Instruktionen werden in das neue Quellenprogramm übernommen. Die gleiche Anzahl von Instruktionen des alten Quellenprogramms wird vom Leser 1 überlesen.
Grundsätzlich gilt, daß Instruktionen, die das Zeichen "Apostroph" enthalten, bei keiner Korrekturmöglichkeit mitgezählt werden.

Bemerkungen:

Auf einem Korrekturstreifen dürfen wahlweise die drei Korrekturmöglichkeiten s, e, k Verwendung finden. Die Korrekturinformationen e und k sind durch das Zeichen ":", die letzte auf dem Lochband in jedem Fall, also auch bei s, mit zwei Doppelpunkten abzuschließen. Die nach s, e, k verwendeten Namen müssen sich in ihrer Reihenfolge nach den Namen des auszubessernden Quellenprogramms richten. Zuerst muß der entsprechende Name vom Leser 2 (Korrekturinformation) gelesen werden, damit über Leser 1 bis zu diesem Namen gelesen (altes Quellenprogramm) und vom Stanzer bis zu diesem Namen dupliziert werden kann (neues Quellenprogramm).
Soll in einem Teil des Quellenprogramms ausgebessert werden, in dem keine Namen verwendet wurden und der letzte aufgetretene Name bereits gelesen ist, besteht die Möglichkeit, nach den Buchstaben s, e, k, eine Oktalzahl mit vorangestelltem + Zeichen anzugeben. Das bewirkt, daß die durch die Oktalzahl angegebene Anzahl von Instruktionen des Quellenprogramms (gezählt von der Instruktion an, die unmittelbar als nächste vom Leser 1 einzulesen ist) dupliziert und anschließend gestrichen, eingeschoben oder korrigiert wird.

4.4.4. Beispiele für die Korrektur

Fehlerhaftes Quellenprogramm

```
                000             wahl
preis           kf              10,40
wahl            k2
                kb
zykl            wrzl
                ef:3.
                340             r1
                150             ende
                ef:6.4.3.
                tab
                211             preis
                240             r3
                mr:3.           preis
                ef:8.7.2.
                000             zykl
ende            020
:
```

Korrekturinformationen

```
e  wahl+1

fakturierung,
                zl
:
k  zykl+2
                170             ende
:
e+2
2
::
```

Ausgebessertes Quellenprogramm

```
                000             wahl
preis           kf              10,40
wahl            k2
                kb
                ;
fakturierung,
                zl
zykl            wrzl
                ef:3.
                340             r1
                170             ende
                af:6.4.3.
                tab
                mr:3.           preis
                af:8.7.2.
                000             zykl
ende            020
;
```

In das fehlerhafte bzw. unvollständige Quellenprogramm wurden eine Zeichenkette eingefügt, ein fehlerhafter Vergleichersprung korrigiert und zwei überflüssige Instruktionen gestrichen.
Nach dem Korrekturlauf ist das neu entstandene Quellenprogramm für eine erneute Übersetzung geeignet.

a.5. Regieanweisung für den Assembler

a.5.1. Speicherplatzbelegung und Hinweise für die Abspeicherung

Der Assembler (Übersetzungsprogramm) belegt 2452 Speicherplätze (oktal). Er ist relativ programmiert und kann unter Beachtung der Tatsachen, daß als Hauptleitadresse der Anfang einer geraden Spur (xx00) gewählt und diese nicht kleiner als 500 sein darf, wahlfrei abgespeichert werden. Weiterhin ist es notwendig, daß sich das Minimalprogramm (B1.1) im Rechner befindet. Vom Assembler werden die Speicherplätze 7436-7537 und 7600-7777 als Arbeitsspeicher verwendet. Außerdem werden die auf den Speicherplätzen 55 und 112 des Minimalprogramms stehenden Ausgabe-Halt-Befehle für Zwischenraum (0320 030.) beim Laden überschrieben. Die Zellen 431-434 werden mit den entsprechenden Sprungbefehlen zu den Programmzyklen des Assemblers gefüllt und durch Betätigung der Tastenkombination B1 bis B4 zur Abarbeitung gebracht (Bedienungsvereinfachung).

Folgender Algorithmus läßt sich für das Laden des Assemblers aufstellen:

1. Eingabe der Hauptleitadresse (HLA)
2. Öffnen der Bahnsperren 40, 100 und 400
   (Ist die HLA kleiner als 2000, müssen die entsprechenden Bahnsperren während des Ladens und während der Übersetzung geöffnet sein.)
3. Abspeichern des Assemblers mit H Kx R
4. Schließen der Bahnsperren 40, 100 und 400

Während des Ladens bzw. nach Abschluß des Ladens können folgende Ausschriften über die Schreibmaschine auftreten:

HLA XXOO!	Als Hauptleitadresse wurde nicht der Anfang einer geraden Spur gewählt. Das Laden muß unter Vorgabe einer solchen HLA wiederholt werden (Lochstreifen zurücklegen).
bahnsp öffnen anwahl st bestätigen	Die Bahnsperren sind geschlossen. Nachdem sie geöffnet wurden, sind nur die Taste des entsprechenden Eingabekanals zu betätigen und die Starttaste zu drücken. Dadurch wird das Laden fortgesetzt (Lochstreifen bleibt im Leser).
ok	Der Ladevorgang ist ordnungsgemäß abgeschlossen.
fe	Die Abspeicherung des Assemblers erfolgte fehlerhaft. (Wenn nach erneuter Eingabe des Minimalprogramms und des Assemblers wiederum fe, ausgeschrieben wird, liegt in der Regel ein Maschinenfehler vor.)

4.5.2. Der erste Übersetzungslauf

Für eventuelle Fehlerausschriften ist ein Tabulatorstopp (linker Randsteller + 20) und für die wahlfreie Ausgabe des Adreßbuches der Schalter 'BS' zu setzen. Anschließend ist das Quellenprogramm in einen der Leser einzulegen und die Tastenkombination H G R B1 zu betätigen. Der Rechner geht danach in den Eingabe-Zustand. Durch die Eingabe einer '1' oder '2' über die Tastatur wird dem Assembler mitgeteilt, über welchen Kanal die Ein- bzw. Ausgabe des Quellen- bzw. Objektprogramms erfolgen soll:

1 = Kanal 1

Das Quellenprogramm ist in den Leser 1 einzulegen, das Objektprogramm (BIS-Programm) wird über den Stanzer ausgegeben.

2 = Kanal 2

Das Quellenprogramm ist in den Leser 2 einzulegen, das Objektprogramm (BIS-Programm) wird über die Schreibmaschine ausgegeben.

Besteht das Quellenprogramm aus mehreren Teillochbändern, ruft das jeweilige Schlußzeichen '.' einen unbedingten Stopp mit der Adresse 1 hervor. Durch Betätigung der Starttaste bzw. der Tastenkombination H G R B3 wird die Übersetzung des nächsten Teillochbandes fortgesetzt. Das Schlußzeichen ':' des letzten Teillochbandes ruft einen bedingten Stopp hervor. Wird anschließend die Starttaste gedrückt, so erfolgt die Ausgabe des Adreßbuches über die Schreibmaschine. Ist die Ausgabe des Adreßbuches unerwünscht, kann durch die Tastenkombination H G R B2 der zweite Durchlauf gestartet werden.
Die Ausgabe des Adreßbuches geschieht automatisch, wenn vor dem Start des ersten Übersetzungslaufes nicht 'BS' gesetzt wurde.
Nach der Ausgabe von jeweils 32 Namen des Adreßbuches geht der Rechner in den bedingten Stopp, damit für die Ausgabe weiterer Namen ein neues Blatt eingespannt werden kann.
Ist das Adreßbuch vollständig ausgegeben, erfolgt ein unbedingter Stopp mit der Adresse 11.

4.5.3. Der zweite Übersetzungslauf

Das Quellenprogramm ist für den zweiten Durchlauf in den bereits im ersten Durchlauf benutzten Leser einzulegen. Steht der Rechner im unbedingten Stopp mit der Adresse 11 (nach vollständiger Adressenausgabe), so wird der zweite Durchlauf durch Betätigen der Start-

taste eingeleitet. Andernfalls ist die Tastenkombination H G R B2 zu drücken.
Der Assembler übersetzt die Instruktionen des Quellenprogramms nacheinander und gibt die einzelnen Befehle des BIS-Programms fortlaufend über den Stanzer oder das Schreibwerk aus.
Nach beendeter Übersetzung geht der Rechner in einen unbedingten Stopp mit der Adresse 22. Durch Betätigen der Starttaste bzw. der Tastenkombination H G R B1 kann zu einem erneuten ersten Durchlauf übergegangen werden.
Das Schlußzeichen "." bei Teillochbändern ruft einen unbedingten Stopp mit der Adresse 2 hervor. Mit Hilfe der Starttaste bzw. der Tastenkombination H G R B4 wird die Übersetzung des nächsten Teillochbandes hervorgerufen.

4.5.4. Beispiel für eine vollständige Übersetzung

Das Quellenprogramm "Fakturierung 1" soll nach erfolgter Ausbesserung (vgl. Pkt. 4.4.4.) übersetzt werden:

```
                000             wahl
preis           kf              10,40
wahl            k2
                kb
                ,
fakturierung,
                zl
zykl            wrzl
                ef:3.
                340             r1
                170             ende
zeile           ef:6.4.3.
                tab
                mr:3.           preis
                ef:8.7.2.
                000             zykl
ende            020
                :
```

Der durch das Ablochen entstandene Quellenprogrammstreifen wird in den Leser 2 gelegt (dadurch erfolgt die Ausgabe des BIS-Programms über die Schreibmaschine), und nach dem Setzen von "BIS" wird der erste Übersetzungslauf mit H G R B 12 gestartet. Nach der Ausschrift iz 0017
bz 0035
geht der Rechner in den bedingten Stopp. Hierbei gibt iz 0017 den Stand des Instruktionszählers (PROSA-Instruktionen) an und bz 0035 den Stand des Befehlszählers (übersetzte Befehle des BIS-Programms). Der unterschiedliche Stand von iz und bz beruht auf der Tatsache, daß eine Makroinstruktion des Programmiersystems PROSA entsprechend dem Übersetzungsverhältnis mehrere Befehle im BIS-Programm ergibt.
Durch die Betätigung der Starttaste wird die Ausgabe des Adreßbuches eingeleitet.

ende	0035
preis	0001
wahl	0002
zykl	0022

Im Adreßbuch werden den symbolischen Adressen des Quellenprogramms numerische Maschinenadressen zugeordnet.
Der symbolischen Adresse "ende" entspricht die relative numerische Maschinenadresse "0035" usw.
Der Assembler sortiert vor der Ausgabe des Adreßbuches die symbolischen Adressen in alphabetischer Reihenfolge. Nach vollständiger Ausgabe des Adreßbuches geht der Rechner in einen unbedingten Stopp. Damit ist der erste Übersetzungslauf beendet.
Durch Betätigung der Starttaste (wenn Adreßbuchausgabe erfolgt) bzw. der Tastenkombination H G R B 2 wird mit dem erneut in den Leser 2 gelegten Quellenprogramm der zweite Übersetzungslauf gestartet.
Zunächst wird über die Schreibmaschine ausgegeben:

num. codierung iz bz fehler

Unter "num. codierung" erfolgt die Ausschrift der einzelnen Befehle und Konstanten des übersetzten BIS-Programms.
Sollten beim Aufstellen bzw. Ablochen des Quellenprogramms syntaktische Fehler entstanden sein, so werden diese unter "iz bz fehler" angezeigt (vgl. Pkt. 4.5.5.).
Anschließend erfolgt die instruktionsweise Übersetzung des Quellenprogramms mit sofortiger Ausschrift (bzw. Ausstanzen) der übersetzten Befehle des Objektprogramms (BIS-Programm):

010002000.	000670030.
510,40	001720030.
004560030.	002720030.
001170030.	0000037572025.
002720030.	007773340.
000460030.	010036170.
000610030.	0031037555025.
001220030.	001550030.
001630030.	010001211.
001440030.	007774240.
001110030.	003000754325.
000510030.	0041627555025.
000450030.	010004000.
001110030.	010000020.
001440030.	/
001050030.	iz 0017
	bz 0035

Der Schrägstrich zeigt das Ende der Übersetzung an. Anschließend erfolgt der nochmalige Ausdruck des Instruktions- und Befehlszählerstandes. Stimmen die entsprechenden Zähler nach dem ersten und zweiten Durchlauf nicht überein, so enthalten das Quellenprogramm und das übersetzte Objektprogramm noch Fehler.

a.5.5. **Fehlerausschriften bei der Übersetzung**

Erkennt der Assembler während der Übersetzung syntaktische Fehler, so erfolgt unabhängig von angewählten Kanal eine Fehlerausschrift (vgl. Pkt. 4.5.4.). Je nach Art des Fehlers sind diese Ausschriften verschieden. Sie lassen sich den Fehlerursachen entsprechend in folgende Gruppen einteilen:

1. **Fehler bei der Adressierung**

a) Im Operandenteil wurde eine symbolische Adresse verwendet, die in keinem Namenteil einer Instruktion auftaucht:

-Name fehlt-
Ausschrift - nf

b) Im Operandenteil wurde eine symbolische Adresse verwendet, die im Namenteil von zwei oder mehreren Instruktionen auftaucht:

-Name mehrfach definiert-
Ausschrift - nmd

c) Der zur Kennzeichnung einer Instruktion gebildete Name wurde nicht den Regeln entsprechend definiert

-verbotener Name-
Ausschrift - vbn

d) Bei Maschinenbefehlen, die im Quellenprogramm zur Anwendung gelangten, wurde die Adresse AC angegeben:

-Adresse-
Ausschrift - adr

e) Im Kennzeichenteil steht ein nicht den Regeln entsprechendes Kennzeichen (z.B. ist bei einem Befehl eine 1 oder der Kennzeichenteil belegt, obwohl er leer sein muß):

-Kennzeichen-
Ausschrift - ks

2. Fehler im Operationsteil

a) Im Operationsteil wurde eine Zeichenkombination verwendet, die in der Befehlsliste nicht definiert ist:

-Operation-
Ausschrift - op

b) Im Operationsteil ist ein nicht zugelassenes Zeichen enthalten:

-verbotenes Zeichen-
Ausschrift - vbz

c) Im Operationsteil wurde eine falsche Parameterangabe programmiert (z.B. mehr als 3 Parameter angegeben):

-Parameter-
Ausschrift - par

3. Sonstige Fehler

Beim Laden des Assemblers mit der kleinsten Leitadresse 500 (oktal) können maximal 1114 (dezimal) verschiedene Namen benutzt werden. - Die obere Grenze des Adreßbuches (Speicherplatz 7434) wurde überschritten:

-Adreßbuch zu groß-
Ausschrift - adrb zg

Bis auf die Fälle

- Name mehrfach definiert,
  verbotener Name und Adressbuch zu groß -

wird vor der Fehlerausschrift noch die Nummer der fehlerhaften Instruktion (Stand iz) und die des Maschinenbefehls, der sich aus der falschen Instruktion ergibt (Stand bz), ausgeschrieben.
Dies erleichtert die Fehlersuche und die Entscheidung über die Art der Korrektur.

a.6. **Befehlsliste**

1. Unterprogrammsprünge für Rechen- und Ausgabeoperationen

Symb.-Code	Bedeutung und Übersetzung	Variante
a) für Festkommazahlen		
M:q.	Multiplikation - Festkomma q000 7542 025	1
D:q.	Division - Festkomma q000 7540 025	1
MR:q.	Multiplikation - Festkomma mit Rückspeicherung q000 7543 025	1
DR:q.	Division - Festkomma mit Rückspeicherung q000 7541 025	1
AF:e.v.n.	Ausgabe - Festkommazahl verkürzt ohne Rundung evn7555 025	1
AFR:e.v.n.	Ausgabe - Festkommazahl verkürzt mit Rundung evn7556 025	1
AF:v.n.	Ausgabe - Festkommazahl vn7557 025	1
b) für Gleitkommazahlen		
ADG	Addition - Gleitkomma 7552 025	1
SBG	Subtraktion - Gleitkomma 7550 025	1
ADGR	Addition - Gleitkomma mit Rückspeicherung 7553 025	1
SBGR	Subtraktion - Gleitkomma mit Rückspeicherung 7551 025	1

Symb.-Code	Bedeutung und Übersetzung	Variante
MG	Multiplikation - Gleitkomma 7546 025	1
DG	Division - Gleitkomma 7544 025	1
MGR	Multiplikation - Gleitkomma mit Rückspeicherung 7547 025	1
DGR	Division - Gleitkomma mit Rückspeicherung 7545 025	1
AG:v.n.	Ausgabe - Gleitkommazahl ohne Rundung vn7560 025	1
AGR:v.n.	Ausgabe - Gleitkommazahl mit Rundung vn7561 025	1
	Ausgabe - Gleitkommazahl halblogarithmisch 7562 025	1

c) für sonstige Darstellungsarten von Zahlworten

ATET:e.v.n.	Ausgabe - Tetraden evn7554 025	1
ATRI	Ausgabe - Triaden 0276 025	1
AADR	Ausgabe - Adreßteil 0100 025	1
ABEF	Ausgabe - Befehlsteil 0053 025	1

2. Festkomma- und Adresseinsaddition und -subtraktion

ADF1	Addition - Festkommaeins 13 210	2

SAPR-Code	Bedeutung und Übersetzung	Variante
ADA1	Addition - Adressenteil 25 210	2
SNF1	Subtraktion - Festkommateil 13 220	2
SDA1	Subtraktion - Adressenteil 25 220	2

3. Unterprogrammaufrufe für Eingabeoperationen

a) für Festkommazahlen

EF	Eingabe - Festkommazahl 7573 025	3
EF:k.	Eingabe - Festkommazahl mit Kommainformation k7572 025	3
EEF	Eingabe - Satz - Festkommazahlen 7565 025	3
EEF:k.	Eingabe - Satz - Festkommazahlen mit Kommainformation k7564 025	3

b) für Gleitkommazahlen

EG	Eingabe - Gleitkommazahl 7574 025	3
EGL	Eingabe - Gleitkommazahl halblogarithmisch 7575 025	3
EEG	Eingabe - Satz - Gleitkommazahlen 7566 025	3
EEGL	Eingabe - Satz - Gleitkommazahlen halblogarithmisch 7567 025	3

Symb.-Code	Bedeutung und Übersetzung	Variante
c) für sonstige Darstellungsarten von Zahlworten		
ETET	Eingabe - Tetraden 7570 025	3
EWOR	Eingabe - Wort mit Kennzeichen 0262 025	3
EADR	Eingabe - Adresse 0316 025	3
EPRO	Eingabe - Programm 0263 025	3
4. Anwahl- und Schreibwerkoperationen		
K0	Anwahl - Kanal 0 4700 030	4
K1	Anwahl - Kanal 1 4600 030	4
K2	Anwahl - Kanal 2 4500 030	4
K3	Anwahl - Kanal 3 4400 030	4
WRZL	Wagenrücklauf mit Zeilenschaltung 2720 030	4
WR	Wagenrücklauf 1120 030	4
ZL	Zeilenschaltung 1720 030	4
GB	Umschaltung Großbuchstabe 0570 030	4
KB	Umschaltung Kleinbuchstabe 1170 030	4
ZWR	Zwischenraum 0320 030	4

Symb.-Code	Bedeutung und Übersetzung	Variante
TAB	Tabulatorsprung 1550 030	

5. Definitionen

KF	Konstante - Festkommazahl K 1 = 5	
KG	Konstante - Gleitkommazahl K 1 = 3	
KGL	Konstante - Gleitkommazahl halblogarithmisch K 1 = 4	
KTET	Konstante - Tetraden K 1 = 6	
KTRI	Konstante - Triaden K 1 = 0 K 2 = 0	
B	Bereich 20 Adresse.	
BN	Bereich auf Null löschen 0.	

werden, daß das Programm mehrmals nacheinander abgearbeitet werden kann, ohne daß es erneut in den Rechner eingegeben werden muß.

alte Anfangsadresse: >a< = 2. Leitadresse
neue Anfangsadresse: >b< = 3. Leitadresse
alte Endadresse : >a< = 4. Leitadresse

2. PROSA - Quellenprogramm

Name	Operation	Parameter	Operand	KZ	Bemerkungen
	211		kon1		
	240		lies		
	211		kon2		
	240		trans		
lies	bn		1		Speicherplätze für
trans	bn		1		Lese- u. Transportbefehl
	ada1		lies		
	330		kon3		
	ada1		trans		
	140		lies		
	020				
kon1	211		0000	2	Konstanten für
kon2	240		0000	3	Lese- u. Transportbefehl
kon3	211		0000	4	Konstante für Abbruch

.
.

3. Protokoll des ersten Übersetzungslaufes:

iz	0015
bz	0021
kon1	0017
kon2	0020
kon3	0021
lies	0004
trans	0005

4. Protokoll des zweiten Übersetzungslaufes:

```
num. Codierung      iz      bz      fehler
01 0017 211.
01 0004 240.
01 0020 211.
01 0005 240.
0.
0.
01 0004 211.
00 0025 210.
01 0004 240.
01 0021 330.
01 0005 211.
00 0025 210.
01 0005 240.
01 0004 140.
01 0000 020.
02 0000 211.
03 0000 240.
04 0000 211.
/
iz 0015
bz 0021
```

Beispiel 3: Maximum von n Zahlen

1. Aufgabenstellung

Es liegt ein Lochstreifen mit beliebig vielen positiven Festkommazahlen a_i vor. Er ist durch "-1" abgeschlossen. Die Festkommazahlen sollen wortweise eingelesen und das Maximum ermittelt werden. Das Maximum ist über Schreibmaschine auszugeben.

Kommainformation:

a_i v = 3
n = 2

2. PROSA - Quellenprogramm

```
Name    Operation:Parameter  Operand   KZ  Bemerkungen
                000          begin
max             bn           1
begin           k2
                kb
                241          max
zykl1           ef:2.                      Eingabe Wort
                040          druck         Negativsprung
                330          max           zum Druck
                150          zykl2
                000          zykl1
zykl2           240          max
                000          zykl1
druck           af:4.3.2.    max           Ausgabe Maximum
                020
```

3. Protokoll des ersten Übersetzungslaufes:

```
iz      0015
bz      0017
begin   0002
druck   0014
max     0001
zykl1   0005
zykl2   0012
```

a. Protokoll des zweiten Übersetzungslaufes:

```
num. Codierung      iz      bz      Fehler
01 0002 000.
0.
00 4500 030.
00 1170 030.
01 0001 030.
00 00027572 025.
01 0014 040.
01 0001 330.
01 0012 150.
01 0005 000.
01 0001 240.
01 0005 000.
01 0001 211.
00 7773 240.
00 20627555 025.
01 0000 020.
/
iz 0015
bz 0017
```

Beispiel 4: Ausgabe Satz

1. Aufgabenstellung

Im Rechner ist von einer Anfangsadresse > aU < ab auf hintereinanderfolgenden Speicherplätzen ein Satz abgespeichert. Der Satz besteht aus 4 Festkommazahlen. Die einzelnen Worte des Satzes sind mit folgender Stellenzahl vor und nach dem Komma auszugeben:

W1 : 6/0
W2 : 8/2
W3 : 8/2
W4 : 6/3

Das Programm ist als Unterprogramm zu programmieren. Die Satzanfangsadresse (> aU <) wird dem Unterprogramm über Register 1 übermittelt.

2. PROSA - Quellenprogramm

Name	Operation	Parameter	Operand	KZ	Bemerkungen
	240		ende		Sicherung des Rücksprunges
	211		r1		
	210		kon1		
	240		kw1		
	ada1				
	240		kw2		
	ada1				
	240		kw3		
	ada1				
	240		kw4		
	wrzl				
kw1	bn		1		
	af	:6.0.	ac		Druck W1
	tab				
kw2	bn		1		
	af	:8.2.	ac		Druck W2
	tab				
kw3	bn		1		
	af	:8.2.	ac		Druck W3
	tab				
kw4	bn		1		
	af	:6.3.	ac		Druck W4
ende	bn		1		
kon1	ktr1		211		Konstante für Lesebefehl
.					
.					

3. Protokoll des ersten Übersetzungslaufes:

```
iz      0027
bz      0033
ende    0032
kon1    0033
kw1     0013
kw2     0017
kw3     0023
kw4     0027
```

4. Protokoll des zweiten Übersetzungslaufes:

```
num. Codierung      iz      bz      fehler

01 0032 240.
00 7773 211.
01 0033 210.
01 0013 240.
00 0025 210.
01 0017 240.
00 0025 210.
01 0023 240.
00 0025 210.
01 0027 240.
00 2720 030.
0.
00 7777 240.
00 0140 7557 025.
00 1550 030.
0.
00 7773 240.
00 0202 7557 025.
00 1550 030.
0.
00 7773 240.
00 0202 7557 025.
00 1550 030.
0.
00 7773 240.
```

```
num. Codierung     iz     bz     fehler
00 01437557 025.
0.
00 0000 211.
/
iz 0027
bz 0033
```

5.

5.1. Die Geräte der ersten Peripherie

Zu den Geräten der 1. Peripherie gehören:
das alphanumerische ansteuerbare Schreibwerk,
der Lochbandstanzer,
die zwei Lochstreifenleser.
Leser und Stanzer können sowohl im 8-Kanal-R 300-Code als auch im 5-Kanal-Fernschreibcode arbeiten. Durch besondere Ein- und Ausgabeprogramme könnte jedoch theoretisch jeder beliebige Code gelesen oder gestanzt werden. Das Schreibwerk ist durch feste Zuordnung der entsprechenden Ausgabeadressen an die vorgenannten Codesysteme gebunden.

5.1.1. Der Lochbandleser 303 bzw. PS 750

Der Lochbandleser Typ 303 entspricht der Standardausrüstung des C 8205. Dieser Leser arbeitet nach dem Prinzip der mechanischen Abtastung der Lochungen mittels Bürsten. Der Lochstreifen läuft dabei von links nach rechts durch die Abfühleinrichtung des Lesers. Durch Schwenken des Bedienungshebels werden die Andruckrolle und die Abtastbürsten abgehoben. Das Lochband kann eingelegt werden. Die Inbetriebnahme der Leser erfolgt getrennt durch zwei zwischen den Lesern befindliche Druckschalter. Durch Absenken der Abtastbürsten wird der Antriebsmotor des Lesers in Betrieb gesetzt. Der Lochstreifen läuft auf der Transportlochung bei Anwahl des entsprechenden Eingabekanals bis zum ersten zu lesenden Zeichen vor. Der eigentliche Lesevorgang wird durch das jeweilige Eingabeprogramm ausgelöst.
Es können 8-Kanal- und 5-Kanalstreifen verwendet werden (der Rechner muß dabei allerdings auf das entsprechende Codesystem eingestellt sein). Bei eventuellen Korrekturen können Lochstreifen überlappend geklebt werden.
Damit die Lochstreifenleser geschont werden, ist die

Anwendung angeölten Lochbandes empfehlenswert. Die Abtastbürsten und die Kontaktplatte sind bei ständiger Benutzung der Anlage täglich mit einem weichen trockenen Pinsel von Staub zu reinigen. Bei starker Verschmutzung kann etwas Tetrachlorkohlenstoff zu Hilfe genommen werden. Es ist darauf zu achten, daß der Lochstreifen nicht rückwärts durch den Leser gezogen wird. Die Abtastbürsten werden dadurch möglicherweise verbogen und die Betriebssicherheit der Leser gestört.

Der Lochbandleser Typ FS 750 arbeitet mit fotoelektrischer Abtastung. Der Lochstreifen läuft im Gegensatz zum Typ 303 von rechts nach links durch die Abtasteinrichtung. Es wird nach einem Start-Stopp-Prinzip gearbeitet. Der Streifen wird dabei durch einen Elektromagneten festgehalten. Der eigentliche Antrieb erfolgt durch Friktion zwischen der Motorwalze und einer Andruckwalze, die bei geschlossenem Abtastblock mit definiertem Druck den Transport des Lochstreifens bewirkt, wenn der Elektromagnet stromlos ist. Das Lesen wird durch Impulse von der Zentraleinheit gesteuert. Die richtige Lage der abzutastenden Lochung wird durch die Transportlochung, die ebenfalls fotoelektrisch abgetastet wird, selbsttätig gesteuert.
Die besondere Arbeitsweise dieses Lesers verbietet die Verwendung von angeöltem Papier sowie überlappendes Kleben von Lochstreifen. Bei der Korrektur von Lochungen sowie beim Kleben von Lochstreifen muß mit undurchsichtiger Klebefolie gearbeitet werden. Streifen müssen auf Stoß geklebt werden.
Die Erfahrungen bei der praktischen Arbeit mit diesem Leser haben auch gezeigt, daß bei der Verwendung einfacher weißer Lochbänder gelegentlich Lesefehler durch die erhöhte Transparenz dieser Streifen auftreten können. Die Verwendung farbiger Lochbänder ist deshalb günstiger. Es empfiehlt sich auch eine tägliche Reinigung der Abtastbahn mit einem weichen trockenen Pinsel.

5.1.2. Der Lochbandstanzer

Der Lochbandstanzer gestattet die Verwendung von 5-Kanal- bzw. 8-Kanal-Lochstreifen. Der zu stanzende Code wird durch das Ausgabeprogramm bestimmt. Bei Anwahl des entsprechenden Kanals (Kanal 1) in der Funktion Ausgabe schaltet sich der Motor bei Ausgabe von Informationen selbsttätig ein. Ist die Ausgabe abgeschlossen, wird der Motor nach ca. 5 s wieder gestoppt. Bei Lochstreifenriß oder Lochbandende wird dieser Zustand über einen Sicherheitskontakt der Zentraleinheit signalisiert und die Ausgabe unterbrochen. Sie kann bei Bedarf fortgesetzt werden.
Es ist darauf zu achten, daß die Stanzabfälle sicher in den unterhalb des Stanzers angeordneten Abfallkasten gelangen. Der Kasten ist rechtzeitig zu entleeren. Verstopfungen führen zur Beschädigung des Stanzers.
Durch zwei auf der Abdeckplatte des Stanzers angeordnete Funktionstasten kann Streifenvorlauf mit Transportlochung bzw. Streifenvorlauf mit Transport- und Irrungslochung (alle 8 Kanäle gelocht) erzeugt werden. Diese Tasten können nur betätigt werden, wenn der Rechner betriebsbereit ist (Lampe "B" leuchtet). Unabhängig von dem automatischen Motor-Start-Stopp kann der Stanzer über die Taste "LS" auf der Tastatur-Anzeigeeinheit auf Dauerbetriebsbereitschaft geschaltet werden.

.... = Transportlochung/Streifenvorlauf

= Transportlochung mit Irrungslochung

5.1.3. Das Schreibwerk

Das Schreibwerk hat beim C 8205 zwei Funktionen, es kann sowohl Eingabe- als auch Ausgabefunktion übernehmen.

Ausgabefunktion

Die Ausgabe erfolgt, wenn es das Ausgabeprogramm er-

laubt, mit einer Geschwindigkeit von maximal 12 Zeichen/s. Alle Funktionen der Schreibmaschine außer Farbbandumschaltung können durch Programm angesteuert werden. Die Ausgabe erfolgt über Kanal 2.

Randeinstellung:
Bei der Randeinstellung ist zu beachten, daß am rechten Anschlag automatisch Wagenrücklauf-Zeilenschaltung ausgelöst wird. Der rechte Randeinsteller kann deshalb normalerweise bei eingeschaltetem Schreibwerkmotor nicht verstellt werden. Wird der Versuch gemacht, so werden durch den ausgelösten Wagenrücklauf beide Anschläge zusammengeschoben und das Schreibwerk blockiert. Der linke Randeinsteller kann durch Niederdrücken der Randverstellertaste (senkrechter Strich) und Betätigung der Wagenauslösetaste (rechts außen auf dem Wagen) in die gewünschte Stellung gebracht werden. Auch hier ist zu vermeiden, den linken Randversteller in die Nähe des rechten Anschlags zu bringen. Alle Tasten sind durch geeignete Symbole eindeutig gekennzeichnet.
Das Schreibwerk kann wahlweise durch den Rechner bzw. manuell über Tastatur betätigt werden. Die Umschaltung erfolgt über die Taste "SW" der Anzeige-Tastatureinheit. Über diese Taste kann die Ausgabe durch den Rechner augenblicklich gestoppt werden (z.B. bei Blattende). Die Ausgabe kann an der gleichen Stelle durch Ausrasten der Taste wieder fortgesetzt werden. Der gleiche Effekt kann auch durch Betätigung der Taste "H" erreicht werden. Hierbei ergibt sich jedoch eine gewisse Verzögerung bis zum Stillstand.

Eingabefunktion

Die Eingabe über die Schreibwerktastatur erfolgt über Kanal 0 und zwar, wenn auf der Tastatur-Anzeigeeinheit die Tastenkombination "T/SW" auf "SW" geschaltet wurde. Dieser Zustand wird durch die Signallampe "SW" angezeigt. Die Eingabe über Schreibwerk hat den Vorteil der gleichzeitigen visuellen Kontrolle und des größeren Zeichenvorrats.

5.1.4. Tastatur und Anzeigeeinheit

Wir unterscheiden bei den Tasten und Schaltern der Tastatur einzelne Funktionsgruppen:

Eingriffstasten
Eingabetasten
Anwahltasten
Selektoren
Sperrschalter

Die obere Reihe zeigt Betriebsbereitschaft und den Arbeitszustand des Rechners an. Die untere Reihe gibt Auskunft über den Stand des Befehlsregisters und zeigt den Zustand der Vergleicher an.

5.1.4.1. Tastatur

Vgl. Abb. 8a, Seite 116.

Eingriffstasten

Die Schalter "H/R" sind gegeneinander verriegelt ("H/R" = Halt/Rechnen). Bei der Abarbeitung von Maschinenbefehlen wird der Rechner durch Betätigung der Taste "H" nach Abarbeitung der nächsten Phase III (vgl. Befehlsschleife) gestoppt. Die Arbeit kann durch Drücken der Taste "R" fortgesetzt werden.
Die Tasten "E", "M", "G", "Ü" und der Schalter "T-SW" können nur im Zustand "H" betätigt werden.

"E" = Einzelschritt:
Mit Hilfe dieser Taste kann eine Befehlsfolge in Einzelschritten abgearbeitet werden. Durch Anschlag dieser Taste wird die Befehlsschleife einmal durchlaufen.

"G" = Generallöschen:
Durch Betätigung dieser Taste wird das Befehlsregister gelöscht, die Zustände "FÜ", "BS", "US"

Tastatur des C 8205

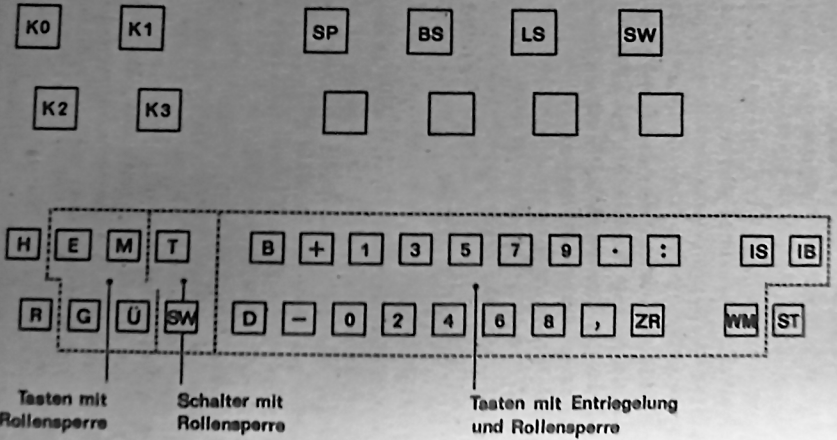


Abb. 8a

beseitigt und der Kanal β (Eingabe über Tastatur) angewählt. Befehlszähler- und AC-Inhalt werden nicht beeinflußt.

"M" = Manuell:
Bei gedrückter Taste "M" und Anschlagen der Taste "E" wird die Befehlsentnahme in Phase II der Befehlsschleife verhindert. Der Inhalt des Operationsregisters wird disjunktiv mit der Operation BAZ = 025. (Sprung mit Rückkehrbefehl) verknüpft. Es erfolgt ein Sprung zur Adresse β. (Auf diese Weise wird z.B. das kleine Eingabeprogramm durch M, L, E, M auf der Adresse 0002 erreicht und angesprungen.)

"Ü" = Übertrag:
Durch Betätigung dieser Taste wird der Inhalt des AC in die Zelle mit der Adresse 7777_8 transportiert.

"T/SW" = Umschaltung zwischen Tastatur und Schreibwerktastatur:
Der jeweilige Betriebszustand wird durch Signallampen angezeigt.

"ST" = Taste "Start":
Diese Taste kann nur im Zustand "US" und "BS" betätigt werden. Sie ist sonst elektronisch verriegelt und wirkungslos. Steht der Automat in einem der genannten Betriebszustände, so erfolgt durch Niederdrücken dieser Taste eine disjunktive Verknüpfung des Inhalts des Operationsregisters mit der Operation BAE = 240.

Eingabetasten

0...9 = Ziffern 0...9
\- = Vorzeichen Minus
\- = Sonderzeichen (z.B. für die Eingabe von Externbefehlen) Dieses Zeichen darf nicht

im Zusammenhang mit der Dezimaleingabe benutzt werden!

, = Komma - wird, wenn nicht erforderlich, überlesen.

. = Punkt - wird im Fernschreib- und R 300-Code als Wortmarke verwendet.

: = Doppelpunkt - wird als Satz- bzw. Blockmarke verwendet.

ZR = Zwischenraum - Korrekturzeichen, löscht AC-Inhalt bei Eingabe.

WM = Wortmarke - nur im R 300-Code erlaubt.

IS = Irrung Satz - stellt den Abspeicherbefehl bei Satzeingabe zurück.

IB = Irrung Block - darf nur im R 300-Code benutzt werden.

Die elektromechanische Sperrung dieser Tastengruppe ist nur dann aufgehoben, wenn folgende Voraussetzungen gegeben sind:

- Kanal 0 muß angewählt sein,
- Eingabe-Halt-Befehl muß im Befehlsregister stehen,
- Schalter "T-SW" muß in Stellung "T" stehen.

Die Eingabe über Tastatur ist der Eingabe über LB-Leser gleichberechtigt.

Anwahltasten, Selektoren

K0 = manuelle Anwahl von Kanal 0, Tastatur bzw. Schreibwerktastatur

K1 = manuelle Anwahl von Kanal 1, Leser 1 bei Eingabe oder Stanzer bei Ausgabe

K2 = manuelle Anwahl von Kanal 2, Leser 2 bei Eingabe oder Schreibwerk bei Ausgabe

SP = Selektor, kann durch Befehl abgefragt werden (Programmverzweigung)

BS = Selektor, kann durch Befehl "Bedingter Stopp" zur Programmverzweigung benutzt werden

Tasten "LS" und "SW"

Mit der Taste "LS" kann der Motor des Stanzers in Dauerbetrieb versetzt werden. (Dadurch wird ein schnellerer Start des Stanzers erreicht, wenn fortlaufend gestanzt werden muß.) Bei nicht gesetzten Schalter "LS" erfolgt der Start des Stanzermotors durch eine Automatik.
Die Taste "SW" ermöglicht die Umschaltung des Schreibwerks von der Ansteuerung durch die Zentraleinheit auf manuelle Ansteuerung über die Tastatur des Schreibwerks. Im Normalfall ist die Tastatur für manuelle Betätigung gesperrt.

5.1.4.2. **Lampenfeld**

Vgl. Abb. 8b, Seite 120

Die Lampen der Anzeigeeinheit sind folgendermaßen eingeteilt:

Die obere Reihe zeigt die Betriebsbereitschaft und den Arbeitszustand des Rechners an.

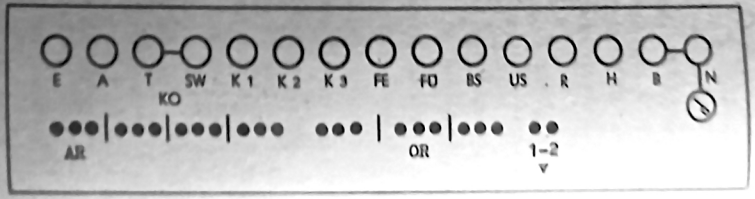
H = Halt (rot)
R = Rechnen (grün)
US = Unbedingter Stopp (rot)
BS = Bedingter Stopp (rot)
FÜ = Überlauf (rot)
FE = Fehler bei Eingabe (rot)
E = Eingabe (weiß)
A = Ausgabe (weiß)

Die untere Reihe gibt Auskunft über den Stand des Befehlsregisters und zeigt den Zustand der Vergleiche an.

Diese Lampen ermöglichen in bitweiser Darstellung das Ablesen von Adreß- und Operationsregister.

	Adreßregister				Operationsregister		
Lampe	○○○	○○○	○○○	○○○	○○○	○○○	○○○
Gewicht	421	421	421	421	421	421	421
Wertigkeit (oktal)	1 000	100	10	1	B	A	Z

Bedeutung der Befehlsregisteranzeige



Lampen der Anzeigeeinheit

Abb. 31

5.2. Inbetriebnahme des Rechners

Die Inbetriebnahme des Rechners erfolgt durch Betätigung des Schlüsselschalters auf der rechten Seite der Anzeigeeinheit. Die Lampe "N" muß dabei aufleuchten. Bei Betätigung der Taste "G" muß die Lampe "B" im gleichen Takt blinken. Nach ca. 5 Minuten muß dieses Blinken in ein Dauerleuchten übergehen (Trommel ist angelaufen). Hört das Blinken der Lampe "B" beim Niederdrücken der Taste "G" nicht auf, so liegt eine Störung vor (z.B. läuft die Trommel nicht an, wenn die Temperatur des Raumes außerhalb der Toleranzgrenze liegt). Nach Ablauf von ca. 25 Minuten hat die Trommel des Rechners ihre Nenndrehzahl erreicht. Die Lampe "B" leuchtet dauernd.

Achtung: Wird unbeabsichtigt die Stromzufuhr unterbrochen, sind bis zur erneuten Betriebsbereitschaft des Rechners wiederum 25 Minuten erforderlich!

5.3. Eingabe des Systems

Die Eingabe des Systems ist im allgemeinen nur erforderlich, wenn der Rechner vorher in einem anderen System betrieben wurde oder die Bausteine des Systems durch Überschreiben zerstört worden sind.

5.3.1. Eingabe des Minimalprogramms

Das Minimaleingabeprogramm ist ein dual notiertes und bitweise gelochtes Programm. Es liegt in Form eines Lochstreifens vor. Die Eingabe erfolgt durch das kleine Eingabeprogramm, das fest auf der Spur Null gespeichert ist. Zur Eingabe des Minimalprogramms müssen folgende Operationen ausgeführt werden:

- Öffnen der Bahnsperren 40, 100, 200, 400 und 1000
- Lochstreifen auf Leser 1 oder 2
- Betätigung der Tasten H G M+R E M+K_x R , K_x = Kanal 1 oder 2

Das Minimalprogramm wird auf den Adressen 0000...0437_8 fortlaufend fest abgespeichert.

5.3.2. Eingabe der Bausteine

Sie erfolgt, nachdem der entsprechende Lochstreifen in Leser 1 oder 2 eingelegt wurde, durch folgende Operationen: H G R 21 Adresse. 22 Adresse. H K_x R
Die Adresse muß mittels der Kennzeichen K_2 = 1 und K_2 = 2 als erste und zweite Leitadresse eingegeben werden. K_x bestimmt dabei über die jeweilige manuelle Kanalanwahltaste den gewünschten Kanal (1 bzw. 2).
Wie bereits an anderer Stelle erwähnt, sind die Bausteine des BIS zeitoptimal aufgebaut. Ihre Abspeicherung ist deshalb nur mit Einschränkungen frei wählbar. Für einige Bausteine gilt, daß sie vom Anfang einer Spur an abgespeichert werden müssen. Die Bausteine 4.2 und 4.3 müssen sogar, wenn sie einwandfrei funktionieren sollen, von einer Zelle mit der Adresse xx00 an abgespeichert werden (als Spuranfang ist bei allen Adressen xx00 bzw. xx40 gegeben).

5.3.3. Eingabe von Programmen (Anwenderprogramme)

Für die Eingabe relativ programmierter Programme muß über Kennzeichen analog zur Bausteineingabe die Hauptleitadresse eingegeben werden. Die Eingabe einer 2. ... 9. Leitadresse ist nur dann erforderlich, wenn sie für die Funktion des jeweiligen Programms vorgesehen ist (vgl. Pkt. 3.3.1.2. - Eingabekennzeichen).

Folgende Informationen müssen dem Rechner zur Programmeingabe mitgeteilt werden: H G R 2 1 Adresse. H K_x R
(K_x = Kanal 1 oder 2)

5.4. Eingabe über Tastatur bzw. Schreibwerktastatur

Wie bereits gesagt, sind alle Eingabegeräte gleichberechtigt. Die Wahl des entsprechenden Mediums ist, abgesehen von dem begrenzten Zeichenvorrat der Tastatur, frei wählbar. So können z.B. auch die Hauptleitadresse oder andere Informationen über den Leser eingegeben werden. Kurze Programme bzw. Programmänderungen können über Tastatur bzw. Schreibwerktastatur erfolgen.

Tastatur der Schreibmaschine

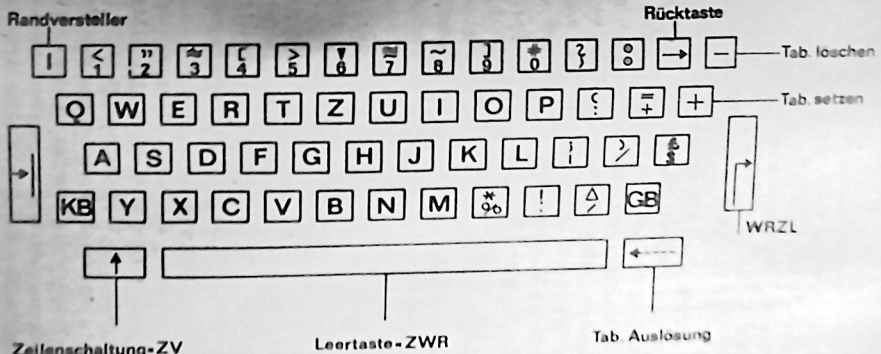


Abb. 9

Die Eingabe über Tastatur wird vorrangig zur Eingabe der Hauptleitadresse (wenn nicht bereits auf LB abgelocht) sowie weiterer Adressen, zum Ansprung der Serviceprogramme, zum Programmstart und zur Programmänderung benutzt. Das gilt besonders dann, wenn Schreibwerk und Leser mit anderen Aufgaben belegt sind.

Bei der Eingabe über Schreibwerk (Schalter "SW" gesetzt) muß beachtet werden, daß beispielsweise bei Betätigung von WRZL oder ZR "Satzmarke" und "Irrung Wort" eingegeben werden. Die Auslösung des automatischen Wagenrücklaufs mit Zeilenschaltung am rechten Anschlag ist ebenfalls zu vermeiden. Der Wagen muß erforderlichenfalls von Hand zurückgeführt werden (vgl. Abb. 9, Seite 123).

5.5. Herstellung der Datenträger

Die Herstellung der Datenträger (Lochbänder) erfolgt auf den entsprechenden Datenerfassungsgeräten:

Orgautomat O 528, daro 1413
C 8031, C 8032, C 8033 (elektromechanischer Locher)
C 8008 (SE 5L im R 300-Code)
Buchungsautomaten (K 117 Le) o. ä.

Je nach Einstellung des Rechners wird im 5-Kanal-Fernschreibcode oder im R 300-Code gearbeitet. Im allgemeinen ist die Anwendung des R 300-Codes gebräuchlicher (größerer Zeichenvorrat).
Alle Informationen, die zusammenhängend eingelesen werden sollen, müssen fortlaufend ohne Zwischenraum abgelocht sein (mit Zwischenraum ist nicht die Lochung ZR gemeint, sondern Transportlochung ohne Information). Bei einer Fehllochung (paarige Lochung) oder Transportlochung während der Eingabe geht der Automat in den Zustand FE "Fehler Eingabe". Dagegen wird eine Transportlochung im Vorspann überlesen. Der Lochstreifen läuft bei Anwahl des entsprechenden Kanals bis zum ersten zu lesenden Zeichen vor.

5.5.1. Programmstreifen

Die Ablochung der Programme erfolgt fortlaufend Zeichen für Zeichen. Da bei der Programmeingabe (Eingabe über das Minimalprogramm) die Eingabesteuerung durch Kennzeichen erfolgt, beginnt jede Information mit einem ersten Kennzeichen, und in Abhängigkeit davon folgt bei $K_1 = 0, 1, 2$ ein zweites Kennzeichen. Nach den Kennzeichen steht unmittelbar das erste von Null verschiedene Zeichen. Führende Nullen brauchen (z.B. bei der Adresse) nicht abgelocht zu werden. Jede Instruktion wird mit einer Wortmarke abgeschlossen (Punkt oder Tab.). Nach Eingabe einer Information geht der Automat zur Eingabe eines neuen ersten Kennzeichens über. Die Verwendung von Zeichen, die nicht als erstes Kennzeichen definiert sind, führt zur Eingabe eines neuen ersten Kennzeichens zurück. Zur besseren Übersicht bei der Ablochung von Programmstreifen ist deshalb Wagenrücklauf/Zeilenschaltung nach jeder Instruktion erlaubt. Ebenso kann vor jeder Instruktion zur besseren Gliederung des Lochbandes die Leertaste beliebig oft betätigt werden.

5.5.2. Datenstreifen

Daten werden, je nach Aufbau des Programms, ohne Kennzeichen in üblicher dezimaler Darstellung abgelocht. Die Informationen können dabei sowohl wortweise als auch in Form von Informationseinheiten (Sätzen, Blöcken) auftreten. Negative Werte werden durch das Zeichen Minus gekennzeichnet, das an jeder beliebigen Stelle der zu erfassenden Zahl eingegeben werden kann. Die Informationen sind durch die vorgeschriebenen Marken abzuschließen (Punkt, Tab., WRZL, :).

Bsp. 2735.768.9876.555 S 2345.345.3456.666.M S

5.5.3. Korrekturmöglichkeiten bei der Lochbandherstellung

Sowohl bei der Programmstreifenherstellung als auch bei der Datenerfassung gibt es verschiedene Möglichkeiten

der Korrektur.
Bei der Programmstreifenherstellung sind folgende Korrekturen möglich:

- Zeichen falsch

1. Betätigung der Rücktaste am Datenerfassungsgerät und Überlochung des falschen Zeichens mit "Irrung Zeichen". Anschließend das falsche Zeichen neu lochen.

2. Wenn das Zeichen Punkt = Wortmarke noch nicht gelocht wurde, kann durch Betätigung der Leertaste (Lochung ZR) erreicht werden, daß das sich im AC aufbauende Wort gelöscht wird. Anschließend wird das gleiche Wort mit dem 1. Kennzeichen nochmals begonnen.

3. Ist das Zeichen Wortmarke bereits gelocht und wird der Fehler erst jetzt bemerkt, so kann durch Eingabe der Relativadresse der falschen Instruktion und der nachfolgenden Sollinformation erreicht werden, daß die falsche Information durch die Sollinformation überschrieben wird. Diese Art der Korrektur kann an beliebiger Stelle (z.B. am Lochbandende) erfolgen.

Bei der Datenerfassung sind folgende Korrekturmöglichkeiten gegeben:

- Zeichen falsch — Analog zur Programmstreifenherstellung sind die unter 1. und 2. beschriebenen Korrekturen möglich.

- Satz falsch — Wird bei der Ablochung von Informationseinheiten in Satzform ein Fehler bemerkt und ist das Zeichen Satzmarke noch nicht gelocht,

so können die unter 1. und 2. genannten Korrekturmöglichkeiten angewendet werden.
Außerdem kann durch Ablochung des Zeichens "Irrung Satz" der Abspeicherbefehl für die Satzeingabe auf die Anfangsadresse des Eingabebereiches zurückgestellt und der Satz erneut abgelocht werden.

5.6. Korrektur des Inhalts von Speicherplätzen

Im Zusammenhang mit der Programmtestung ergibt sich häufig die Notwendigkeit, bei der Erkennung von Fehlern den Inhalt einzelner Speicherplätze zu überschreiben. Dabei ist wie folgt zu verfahren:

1. Eingabe der Hauptleitadresse 21 des Programms
2. Eingabe der Relativadresse 20 des Speicherplatzes
3. Eingabe der Sollinformation mit Kennzeichen
   (z.B. 01 $a_1a_2a_3a_4$ BAZ.)

Prinzipiell ist es dabei gleich, ob die Korrekturen über Tastatur, Schreibwerktastatur oder Lochstreifenleser vorgenommen werden.

5.7. Programmstart

Der Programmstart erfolgt mittels eines Sofortbefehls, und zwar durch einen Sprungbefehl zur Startadresse des Programms.

Z.B. 10 $a_1a_2a_3a_4$ 000.

Beginnt das Programm auf der Hauptleitadresse, kann auch ein verkürzter Ansprung verwendet werden, z.B. 11.

Das entspricht folgender Darstellung: 11 0000 000.

Da führende Nullen nicht eingegeben werden, wird über das zweite Kennzeichen $K_2 = 1$ eine Adressenmodifikation mit der Hauptleitadresse durchgeführt. Somit ergibt sich intern die gleiche Wirkung. 11 0000 000. ist identisch mit 11.

5.8. Eingriffsmöglichkeiten beim Auftreten bestimmter Betriebszustände - Beseitigung der Betriebszustände "US", "BS", "FE" und "PÜ"

Der Zustand "US" kann durch folgende manuelle Eingriffe geändert werden:

1.	Schalter "H/R" in Stellung "H" Taste "G" anschlagen		Programm unterbrochen
bzw. 2.	Taste "ST"	anschlagen	Unbedingter Stopp übergangen - Speicherplatz des zur Adresse gehörenden US-Befehls kann geändert werden
bzw. 3.	Schalter "H/R" in Stellung "H" Taste "Ü" anschlagen Schalter "H/R" in Stellung "R"		Unbedingter Stopp übergangen - Speicherplatz 7777_8 kann geändert werden
bzw. 4.	Schalter "H/R" in Stellung "H" Taste "M" anschlagen Schalter "H/R" in Stellung "R"		Der nächste Befehl wird abgearbeitet

Für den Zustand "BS" gelten die gleichen Eingriffsmöglichkeiten. Zusätzlich kann der Schalter "BS" gelöst werden. Die Abarbeitung wird beim nächstfolgenden Befehl fortgesetzt.

Der Zustand "PÜ" kann folgendermaßen beseitigt werden:

1.	Schalter "H/R" in Stellung "H" Taste "G" anschlagen		Programm unterbrochen
2.	Schalter "H/R" in Stellung "H" Taste "E" gedrückt lassen		Überlauf beseitigt und Abarbeitung des Programms fortgesetzt
	Taste "G" anschlagen Taste "E" loslassen Schalter "H/R" in Stellung "R"		< AC > wird gelöscht, d.h., die Information, die den Überlauf verursacht, geht verloren.

Der Zustand "FE" kann nur durch Generellöschen beseitigt werden. Der Programmablauf wird dadurch in jedem Fall

unterbrochen.

- Transport des Befehlszählers in den Akkumulator

Schalter "H/R" in Stellung "H"
Taste "G" anschlagen
Taste "M" gedrückt lassen
Taste "E" anschlagen
Taste "M" loslassen
Taste "E" anschlagen

- Transport des AC-Inhalts nach Zelle 7777_8

Schalter "H/R" in Stellung "H"
Taste "G" anschlagen
Taste "Ü" anschlagen
Taste "E" anschlagen

Der Inhalt der Zelle 7777 kann anschließend über Serviceprogramm durch D3 7777.. (Ausschreiben des Inhalts einer Speicherzelle) ausgeschrieben werden.

5.9. Hinweise zur rationellen Testung von Programmen

Die auf elektronischen Rechnern zur Abarbeitung gelangenden Programme müssen vor ihrer endgültigen Nutzung auf sachliche und strukturelle Richtigkeit geprüft werden. Dieser Vorgang wird im allgemeinen als Test bezeichnet.
Er läßt sich in folgende Etappen untergliedern:

1. Trockentest des Programmablaufplanes
2. Trockentest des geschriebenen Maschinenprogramms
3. Vergleich des geschriebenen Maschinenprogramms mit der beim Ablochen entstandenen Kontrollausschrift
4. Vergleich des geschriebenen Maschinenprogramms mit dem ausgeschriebenen Speicherinhalt
5. Abarbeitung des Programms mit Testdaten

Die Etappen 1 - 3 sind ohne Benutzung des Rechners durchzuführen, während die Etappen 4 und 5 direkt am Rechner vorgenommen werden.

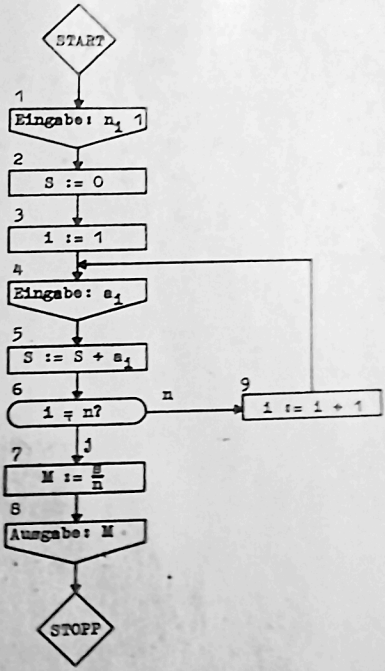
5.9.1. Vorbereitung des Programmtests durch die Trockentests

Die Trockentests stellen eine wichtige Kontrolle der einzelnen Lösungsetappen der Programmierung dar. Der erste Trockentest wird bereits nach der Fertigstellung des PAP durchgeführt.
Seine Aufgabe besteht darin, das Programm auf Vollständigkeit sowie strukturelle und logische Richtigkeit zu überprüfen, z.B.:

- sind alle logischen Kontrollen berücksichtigt,
- sind alle Sonderfälle beachtet,
- sind die Abbruchbedingungen bei Zyklen richtig.

Anhand der Programmieraufgabe "Arithmetisches Mittel von n Zahlen" soll der dafür entwickelte PAP als Beispiel einem Trockentest unterzogen werden:

PAP: Arithmetisches Mittel von n Zahlen



Trockentest

Ausgangsdaten: $n = 5$

$a_1 = 22,\ a_2 = 49,\ a_3 = 81,\ a_4 = 57,\ a_5 = 76,$

1. Durchlauf:
PS1 n = 5, Bereitstellung einer dez. 1
PS2 Summenspeicher S := 0
PS3 Zähler i := 1
PS4 Eing. von $a_1 = 22$
PS5 S := 22
PS6 i := 5? Sprung nach PS9
PS9 i := 1 + 1 = 2 und Sprung nach PS4

2. Durchlauf:
PS4 Eing. von $a_2 = 49$
PS5 S := 22 + 49 = 71
PS6 2 = 5? Sprung nach PS9
PS9 i := 2 + 1 = 3 und Sprung nach PS4

3. Durchlauf:
PS4 Eing. von $a_3 = 81$
PS5 S := 71 + 81 = 152
PS6 3 = 5? Sprung nach PS9
PS9 i := 3 + 1 = 4 und Sprung nach PS4

4. Durchlauf:
PS4 Eing. von $a_4 = 57$
PS5 S := 152 + 57 = 209
PS6 4 = 5? Sprung nach PS9
PS9 i := 4 + 1 = 5 und Sprung nach PS4

5. Durchlauf:
PS4 Eing. von $a_5 = 76$
PS5 S := 209 + 76 = 285
PS6 5 = 5? Sprung nach PS7
PS7 $285 = 57 \cdot 5$
PS8 Ausg. := 57
STOPP

Der Trockentest des geschriebenen Maschinenprogramms läuft prinzipiell nach gleichen Gesichtspunkten ab. Besondere Aufmerksamkeit ist hierbei der Überprüfung der Abbruchbedingungen für die Programmzyklen zu widmen

5.9.2. Festlegung des Testablaufs

Bevor ein abgelochtes Programm am Rechner getestet werden kann, ist es unbedingt erforderlich, die notwendigen Schritte der Testung und die Reihenfolge ihrer Durchführung organisatorisch genau vorzubereiten. Dabei ist für den Testablauf bei Übungen folgender Algorithmus verbindlich:

1. Aufschreiben aller Bausteine und Serviceprogramme, die für den Test benötigt werden.
2. Festlegung der Anfangsadressen für das Abspeichern der Bausteine.
3. Eingabe der Hauptleitadresse und eventuell benötigter Leitadressen und sofortige Kontrolle der entsprechenden Speicherplätze (7721 bis 772x) mit der Tastenkombination D3.
4. Überprüfen des Abspeicherbefehls mit D3 7745..
5. Eingabe des Programms mit H K_x R
6. Ausschreiben der Befehle mit D3 Anfangsadresse. - Endadresse. Ausschreiben von Zahlen mit D4 oder D5 Anfangsadresse. - Endadresse.
7. Korrektur von falschen Speicherplätzen mit Relativadresse, z.B. Befehl Nr. 3 ist falsch -20). Korrigierter Befehl mit Eingabekennzeichen.
8. Ansprung des Programms mit Sofortbefehl
   10 > Programmbeginn < 000.
   Stimmt der Programmbeginn mit der MLA überein, ist der Ansprung auch mit 11. möglich.
9. Ist das Programm durch einen Stoppbefehl (020) beendet worden, muß die rote Lampe "US" als Zeichen der vollständigen Abarbeitung des Programms aufleuchten.
10. Kontrolle des Ergebnisses, wenn nicht vom Programm eine Ausgabe vorgesehen ist, je nach Charakter des Ergebnisses mit D3, D4 oder D5 Anfangsadresse. - Endadresse. (Triaden-, Gleitkomma- oder Festkommadarstellung).

gramm als Grundlage muß immer abgespeichert sein. Alle anderen Bausteine können in frei wählbarer Reihenfolge, ganz nach Bedarf des jeweilig abzuarbeitenden Programms, zusammengesetzt werden.

5.9.4. Adressierungsregeln für die Abspeicherung von Bausteinen

Obwohl die Bausteine, außer dem Minimalprogramm, in der Reihenfolge der Abspeicherung beliebig gewählt werden können, gibt es für die Adressierung einige Regeln, die unbedingt einzuhalten sind. Besonders aus Gründen der zeitoptimalen Arbeitsweise der Bausteine müssen diese am Anfang einer Spur abgespeichert werden. Bei allen Eingabebausteinen und den Bausteinen B 4.2 und B 4.3 wird darüberhinaus gefordert, daß sie am Anfang einer geraden Spur (Adresse xx00) abgespeichert werden. Das Minimalprogramm wird durch die Betätigung der Tastenkombination H G M↓E E M↑K_x R vom KEP eingelesen. Bei sämtlichen anderen Bausteinen muß die Anfangsadresse als Hauptleitadresse und als zweite Leitadresse vorgegeben werden. Soll z.B. ein Baustein ab Adresse 1200 abgespeichert werden, so ist vor Eingabe des Lochstreifens

21 1200.
22 1200.

einzugeben.
Die Angabe der Anfangsadresse als zweite Leitadresse ist notwendig, um die Anschlußzellen des jeweiligen Bausteins nach seiner eigentlichen Abspeicherung mit Sprungbefehlen zum Baustein hin zu versorgen. Hierbei wird die ursprüngliche HLA zerstört, und die richtige Adresse in den Sprungbefehlen wird über das Kennzeichen 2 -2. Leitadresse- realisiert.

Beispiel für das Abspeichern ab Spuranfang einer geraden Spur:

Ein Baustein belegt zum Beispiel Speicherplätze bis 1137. Soll anschließend ein Baustein, für den die Forderung nach Abspeicherung auf den Anfang einer geraden

Spur besteht, abgespeichert werden, muß die Anfangsadresse bei fortlaufender Abspeicherung 1200 lauten. Ansonsten könnte der Baustein schon bei 1140 abgespeichert werden.

5.9.5. Serviceprogramme als Testhilfen

Jeder Baustein stellt prinzipiell ein eigenständiges Unterprogramm dar. Trotzdem gibt es zwischen den Bausteinen Abhängigkeiten. So ist bei der Verwendung einiger Servicebausteine das Vorhandensein eines anderen Bausteins die Voraussetzung für die Arbeit (vgl. dazu Pkt. 5.9.5.1. bis 5.9.5.5.).
Weiterhin ist bei der Programmierung im BIS zu berücksichtigen, daß in allen Bausteinen markierte Einzelbefehle benutzt werden, d.h., durch die Verwendung eines Unterprogramms des BIS wird ein bestehender Vergleicherzustand verändert. Demzufolge können Vergleicher nur vor dem Aufruf vom Unterprogramm des BIS erzeugt und abgefragt werden.

In den drei Servicebausteinen sind die folgenden Programme zusammengefaßt:

B 6.1 Protokollprogramm - Ansprung mit D6, D7
B 6.2 a) Druck des < 7777 > - Ansprung mit D0, D1, D2
b) Bereichsausschreiben - Ansprung mit D3, D4, D5
B 6.3 Vergleichsprogramm - Ansprung mit D0.

5.9.5.1. Bereichsausschreiben

Um ein Programm durch einen Maschinentest auf seine Richtigkeit zu überprüfen, besteht die Grundvoraussetzung darin, daß das Programm so auf dem Speicher angekommen ist, wie es ursprünglich geschrieben und abgelocht wurde. Das kann durch die Benutzung des Bausteins B 6.2 (Bereichsausschreiben) überprüft werden. Mit Hilfe von B 6.2 können Teile von Programmen oder auch ganze Programme, die sich auf dem Speicher be-

finden, ausgeschrieben werden. Mit B 6.2 wird der Inhalt aller Zellen von der Anfangsadresse $a_1a_2a_3a_4$ ab bis zu einer Endadresse $b_1b_2b_3b_4$ ausgedruckt. Bei der spaltenweisen Ausgabe wird die absolute Adresse des jeweiligen Speicherplatzes mit ausgeschrieben. Da die Information auf einem Speicherplatz verschiedenen Charakter haben kann (z.B. Befehl, Maske, Festkommazahl, Gleitkommazahl), unterscheidet man 3 Arten des Bereichsausschreibens:

1. Bereichsausschreiben mit D3
   Die Ausgabe eines Wortes erfolgt als Befehl, wenn die Wortstellen Z_0 bis Z_{11} nicht besetzt sind, andernfalls in Triadendarstellung.

2. Bereichsausschreiben mit D4
   Die Ausgabe eines Wortes erfolgt als Gleitkommazahl, wenn das Bit Z_7 (höchstes Mantissenbit einer Gleitkommazahl) besetzt ist, andernfalls als 11-stellige Oktalzahl.
   Bei Verwendung von D4 muß der Baustein B 4.2 abgespeichert sein.

3. Bereichsausschreiben mit D5
   Die Ausgabe einer Festkommazahl (ganze Dezimalzahl) erfolgt, wenn das letzte Bit Z_{32} nicht besetzt ist, andernfalls als Befehl oder Oktalzahl. Bei Verwendung von D5 muß der Baustein B 4.4 abgespeichert sein.

Beispiel:
Auf den Speicherplätzen 5010 bis 5021 sollen folgende Informationen eingegeben und anschließend kontrolliert werden:

Informationen		Eingabebild
1. Gk-zahl 6,25	Dez→Gk	3 6,25.
2. Befehl 0000020.	Triaden	000000020.
3. Fk-zahl 3000	Dez→Fk	5 3000.
4. Adressensins	Triaden	00 1000.
5. Maske 3777777.	Triaden	00 3777777.
6. Befehl 2000 211.	Triaden	00 2000211.
7. Befehl 2010240.	Triaden	00 2010240.
8. Fk-zahl 1,23	Dez→Fk	5 1,23.
9. Gk-zahl -0,75	Gk → Gk	4-75.0.
10. Gk-zahl 0,075	Gk → Gk	475.-1.

Zellen 5010 bis 5021 mit D3 ausgeschrieben

```
5010 2171 0000 000
5011 0000 020          x
5012 0013 560
5013 0001 000          x
5014 3777 777          x
5015 2000 211          x
5016 2010 240          x
5017 0000 366
5020 6030 0000 000
5021 1663 1463 146
```

Zellen 5010 bis 5021 mit D4 ausgeschrieben

```
5010 6250000 01        x
5011 0000 020
5012 0013 560
5013 0001 000
5014 3777 777
5015 2000 211
5016 2010 240
5017 0000 366
5020 -7500000 00       x
5021 7500000-01        x
```

Zellen 5010 bis 5021 mit D5 ausgeschrieben

5010	1200619520	
5011	8	
5012	3000	x
5013	256	
5014	3777 777	
5015	2000 211	
5016	264272	
5017	123	x
5020	1048576000-	
5021	993211187	

Erläuterungen:

1. Die mit x gekennzeichneten Informationen sind mit der zutreffenden Tastenkombination ausgeschrieben worden.
2. Beim Ausschreiben mit D4 wird zwischen Gleitkommazahl und Befehl unterschieden, deshalb sind die Informationen in Triadendarstellung dort ebenfalls richtig.
3. Beim Ausschreiben mit D5 wird am Bit z_{32} zwischen Festkommazahl und Befehl unterschieden. Befehle mit ungerader letzter Triade im Operationsteil werden deshalb auch richtig ausgeschrieben, die mit gerader letzter Triade im Operationsteil aber nicht.

Schlußfolgerung:

Nur die bewußte Wahl der entsprechenden Tastenkombination verschafft die Gewißheit, ob eine Information richtig bei der Zieladresse abgespeichert wurde.

5.9.5.2. Protokollprogramm

Das Protokollprogramm (B 6.1) kann zum Testen eines Programms benutzt werden. Es arbeitet das vorgegebene Hauptprogramm Befehl für Befehl ab und wird durch D6 oder D7 angesprungen. Bei D6 wird der < AC > als Oktalzahl oder als Gleitkommazahl gedruckt, bei D7 als Oktalzahl oder Festkommazahl. Bei Ansprung des Protokollprogramms durch D6 muß der Baustein B 4.2 Gleitkommaausgabe, bei D7 der Baustein B 4.4 Festkommaausgabe, Variante 1 abgespeichert sein. Nach dem Ansprung des Protokollprogramms durch D6 oder D7 wird durch Eingabe eines weiteren Zeichens entschieden, ob und welche Unterprogramme mit zu protokollieren sind:

0 – alle UP werden protokolliert
1 – UP werden nicht protokolliert
2 – UP des BIS werden nicht protokolliert
3 – alle UP, die durch den Operationsteil 025 angesprungen werden, werden nicht protokolliert

Danach wird die Adresse des ersten zu protokollierenden Befehls angegeben. Die folgende Eingabe von zwei weiteren Adressen legt fest, von welcher Anfangsadresse bis zu welcher Endadresse der < AC > auszudrucken ist. Nachdem bis zu dieser Endadresse protokolliert worden ist, geht der Rechner in US.

Das Eingabebild erhält damit die Gestalt:

D { 6 / 7 } { 0 / 1 / 2 / 3 } a.b.c.

Dabei sind: a – Adresse des ersten zu protokollierenden Befehls
b – Adresse, von der ab die Ausgabe < AC > erfolgt
c – Adresse, bis zu der die Ausgabe < AC > erfolgt

Mit dem Protokollprogramm wird ausgedruckt:

- bei Sprungbefehlen : Zieladresse des Sprungs und < AC > nach Ausführung des Sprungs
- bei Anwahlbefehlen : AWK - K = 1 = Anwahl K1
  K = 2 = Anwahl K2
- bei Eingabebefehlen: EGB (B = Adresse der Eingabebank)
- bei Sprungbefehlen mit Rückkehrabsicht:
  UPa (a = Zieladresse des Sprungbefehls)
- bei Überlauf : a (a = Adresse des Befehls, bei dem der Überlauf erfolgt ist)

Bei Druckbefehlen werden nur Kleinbuchstaben berücksichtigt, und nach jedem Druckbefehl wird der < AC > angegeben.
Nach dem Druck des EGB geht der Rechner in UG.

Das Protokollieren kann mit Hilfe der folgenden Tastenkombination fortgesetzt werden:

H Starttaste E E E E G Ü E G M E M SP setzen R

Die Adresse des nächsten abzuarbeitenden Befehls und der durch den Befehl entstehende < AC > werden in Triadendarstellung ausgegeben.

Im Unterricht kann das Protokollprogramm hauptsächlich für folgende Zwecke zur Anwendung gelangt werden:

1. Anhand eines Protokolls kann besonders schwächeren Lehrlingen unter Anwendung des didaktischen Prinzips der Anschaulichkeit die befehlsweise Abarbeitung eines Programms mit den jeweiligen Zwischen- und Endwerten deutlich vor Augen geführt werden (vgl. bes. Beispiel a).

2. Die Fehlersuche bei fehlerhaften Programmen kann wesentlich vereinfacht werden, u die Fehler optisch sichtbar werden (vgl. Beispiel b - d).

Einschränkend muß bemerkt werden, daß der Zeitaufwand für das Protokollieren eines Programms sehr groß ist.

Deshalb empfiehlt sich der Einsatz nur dann, wenn der Fehler auf einfachere Art nicht auffindbar ist.

Als Beispiel für das Protokollieren wird die Aufgabe Fakturierung 1 verwendet.

Aufgabenstellung: Von einer Anzahl Posten eines Artikels soll aus der Menge M und dem Einzelpreis EP der Preis P der einzelnen Posten errechnet und ausgedruckt werden.
Die Mengenangaben M seien in Dezimaldarstellung auf einem Lochstreifen abgelocht, dessen Ende durch Ø gekennzeichnet ist. Der Einzelpreis EP wird im Programm bereitgestellt.

Druckbild:

M1	P1
M2	P2
.	.
.	.
.	.
Mn	Pn

Kommainformation: Eingabe M mit n = 3
Druck M mit v = 4, n = 3
Druck P mit v = 7, n = 2

Fakturierung 1: Programm

```
3500 3502 000          Sprung über Parameter (EP)
3501      250
3502 4500 030          Anwahl K2
3503 1170 030          KB
3504 3501 211          Lesen EP
3505 7774 240          Transport ins R3
3506 2720 030          WRZL
3507 0003 7572 025     Eingabe Wort - Menge
3510 7773 340          Transport ins R1
3511 3517 170          Sprung bei = zum Stopp
3512 3103 7555 025     Ausgabe M
3513 1550 030          TAB
3514 3000 7543 025     Multiplikation EP * M
3515 4162 7555 025     Ausgabe Preis
3516 3506 000          Zyklussprung
3517 0000 020          Stopp
```

Datenstreifen mit Mengenangaben: 3,75. 1,25. 0.

a) Protokoll zur Fakturierung 1

D733500.3500.3517. Eingabebild zum Beispiel Fakturierung 1

```
  3502     Sprung über Parameter (Zieladresse des Sprungs)
     0     < AC > nach Ausführung des Sprungs
AW 2       Anwahl K2
     0     < AC >

     0     Umschaltung KB mit < AC > (Schreibmaschinenfunktion)
   250 )   Lesebefehl mit < AC >
   250 )   Transportbefehl mit < AC >

   250     WRZL mit < AC > (Schreibmaschinenfunktion)
UP- 7572   UP-Ansprung Eingabe
  3750     < AC > nach Eingabe
  3750 )   Transport mit VOL-Setzen und < AC >
  3750 )   Sprung mit nichterfüllter Sprungbedingung und < AC >
UP  7555   UP-Ansprung Ausgabe
```

```
    3,750   Ausgegebener Wert
     3750   < AC > nach Ausgabe

     3750   TAB mit < AC > (Schreibmaschinenfunktion)
UP   7543   UP-Ansprung Multiplikation
      937   Ergebnis der Multiplikation im AC
UP   7555   UP-Ansprung Ausgabe
     9,37   Ausgegebener Wert
      937   < AC > nach Ausgabe
     3506   Zyklussprung (Zieladresse des Sprungs)
      937   < AC >

      937
UP   7572
     1250
     1250
     1250
UP   7555
     1,250
     1250   } siehe obere Klammer mit 2. Eingabewert
     1250
UP   7543
      312
UP   7555
     3,12
      312
     3506
      312

      312   WRZL mit < AC > (Schreibmaschinenfunktion)
UP   7572   UP-Ansprung Eingabe
        0   < AC > nach Eingabe
        0   Transport mit VGL-Setzen und ≠ AC >
     3517   Sprung mit erfüllter Sprungbedingung
        0   < AC > nach Ausführung des Sprunges
            Ende des Programms und des Protokolls
            (durch Stoppbefehl)
```

b) Protokoll zur Fakturierung 1 bei fehlerhaftem Befehl auf Speicherplatz 3511 - statt 3517 170. (Sprung bei = 0) 3517 150. (Sprung bei > und = 0)

```
D73 3500. 3500. 3517.

   3502
      2000
AW 2
      2000
      2000
       250
       250
       250
UP 7572
      3750
      3750
   3517
      3750
```

Vergleiche Protokoll a

F E H L E R :
Sprung bei > 0 zum Programmende, nicht nur bei = 0, wie in der Aufgabenstellung verlangt

c) Protokoll zur Fakturierung 1 bei fehlerhaftem Befehl auf Speicherplatz 3504 - statt 3501 211. (Einlesen EP) 3515 211. (Einlesen eines Befehls statt EP - demzufolge bei der späteren Multiplikation FÜ)

```
D73 3500. 3500. 3517.

   3502
      3750
AW 2
      3750
      3750
4162 7555 025
4162 7555 025
```

Vergleiche Protokoll a

```
4162 7555 025
UP 7572
      3750

      3750
      3750
UP 7555
    3,750
      3750
      3750
UP 7543
```

F E H L E R :
FÜ, da bei EP mal 41627555025 der Zahlenbereich überschritten wird. FÜ wird nicht protokolliert, da er im UP auftritt.

d) Protokoll zur Fakturierung 1 bei fehlerhaftem Befehl auf Speicherplatz 3504 - statt 3501 211. (Einlesen EP) 0001 211. (falsches Kennzeichen -00- verwendet, deshalb Einlesen einer Konstanten aus dem KEP statt des EP - demzufolge völlig falsche Rechenergebnisse)

D73 3500. 3500. 3517.

```
   3502
        0
AW 2
        0
        0
     8192
     8192
     8192
UP 7572
      3750
      3750
      3750
UP 7555
   3,750
```

F E H L E R :
Falscher Wert, nicht der vorgesehene EP -250-

```
      3750
      3750
UP 7543
     30720
UP 7555
    307,20
     30720
    3506
     30720
     30720
UP 7572
      1250
      1250
      1250
UP 7555
    1,250
      1250
      1250
UP 7543
     10240
UP 7555
    102,40
     10240
    3506
     10240
     10240
UP 7572
         0
         0
    3517
         0
```

Vergleichs Protokoll s

5.9.5.3. Druck von < 7777 >

Dieses Serviceprogramm ist im Baustein 6.2 enthalten. Mit Hilfe der Tastenkombination H G Ü E kann der < AC > nach dem Speicherplatz transportiert werden. Durch anschließendes Drücken von DØ, D1 oder D2 wird der < AC > in

DØ = Triadendarstellung
D1 = Gleitkommazahl
D2 = Festkommazahl

ausgeschrieben.
Dabei wird der < 7777 > nicht zerstört, so daß DØ, D1, D2 in beliebiger Reihenfolge hintereinander gedrückt werden können.
Für D1 ist B 4.2 - Gleitkommaausgabe, für
D2 ist B 4.4 - Festkommaausgabe, Variante 1
vorher abzuspeichern.

Beispiel:

Der beim Beispiel c entstandene FÜ wird durch Ausschrift des < AC > überprüft.

Kontrolle des Inhalts < AC > bei FÜ durch Druck von < 7777 >

```
3,750              FÜ
H G Ü E R D2
2147483648 -       = < AC >
```

5.9.5.4. Vergleichsprogramm

Das Programm B 6.3 kann mit BØ oder einem Sofortsprungbefehl zur Anfangsadresse erreicht werden.
Nach dem Ansprung des Programms ist der Eingabekanal manuell anzuwählen.
Mit Hilfe des Vergleichsprogramms wird eine über Tastatur, Schreibmaschine oder Leser eingegebene Information mit der zugehörigen im Speicher stehenden Information verglichen. Stimmen die Worte nicht überein, werden die Adressen des Speicherplatzes, bei

denen der Unterschied auftrat, die zugehörige Information auf dem Lochstreifen und der Inhalt der Zelle in Triadendarstellung gedruckt.

Beispiel:

Die Aufgabe Fakturierung (vgl. Pkt. 5.9.5.2.) steht mit einem Fehler auf 3504 im Speicher.
Der LS Fakturierung 1 mit Fehlern auf 3510 und 3515 wird über Leser 2 eingegeben.

Beispiel für das Vergleichsprogramm (Fakturierung)

Adr.	Information auf dem LS			Information im Speicher		
3504	0000	3501	211	0000	3515	211
3510	0000	7773	240	0000	7773	340
3515	0000	4162	025	4162	7555	025

Das Vergleichsprogramm speichert die eingelesenen Informationen nicht ab und verarbeitet nur Worte mit Eingabekennzeichen.
Wenn die Hilfsadressen (z.B. HLA) nicht auf dem LS stehen, müssen sie vor der Kanalanwahl eingegeben werden.
Für die Eingabe des LS ist der Leser 2 anzuwählen.

5.9.5.5. Bereichsausstanzen

Mit Hilfe dieses Programms (B 4.1) können Informationen, die sich auf dem Speicher befinden, in einer Form ausgestanzt werden, die für die Eingabe in den Rechner geeignet ist.
Es wird mit DS $a_1\ a_2\ a_3\ a_4$ und $b_1\ b_2\ b_3\ b_4$ angesprungen.
Es wird der Inhalt aller Zellen von $a_1\ a_2\ a_3\ a_4$ bis $b_1\ b_2\ b_3\ b_4$ ausgestanzt (in Triadenform mit Kennzeichenteil 00).

Beispiel : Es soll der Inhalt der Zellen 3015 bis 3042 ausgestanzt werden.

Eingabebild: DS 3015. 3042.

Sollte aus irgendwelchen Gründen die Abarbeitung oder der Test eines Programms abgebrochen werden müssen, so kann es sehr rationell sein, wenn die erreichten Zwischenergebnisse oder der erreichte Stand des Programms gerettet werden. Dies ist durch das Bereichsausstanzen möglich.
Zu beachten ist, daß mit Hilfe des Bereichsausstanzens ausgegebene Programme absolute Adressen besitzen. Wenn das so gewonnene Programm später in den Rechner eingelesen wird (mit HK 1/2 R), kann es nur auf den Speicherplätzen stehen, von welchen aus es gestanzt wurde (als erste Information wird immer die HLA ausgestanzt; in unserem Beispiel also 21 3015.).

Anschlußzellen der Bausteine des EIS

Baustein	Programm	Ansprungzelle	Inhalt
B 2.1	Eingabe einer Gleitkommazahl ohne Kennzeichen	7575	02 336 000.
B 2.1	Eingabe einer Dezimalzahl, Umwandlung intern in Gleitkommazahl	7574	02 335 000.
B 2.1	Eingabe von Dezimalzahlen im Satz ohne Kennzeichen, Umwandlung in Gleitkommazahl	7566	02 315 000. 02 267 000.
B 2.2	Eingabe einer Dezimalzahl ohne Kennzeichen, Umwandlung in Festkommazahl	7573	02 121 000.
B 2.1	Eingabe von Gleitkommazahlen im Satz ohne Kennzeichen	7567	02 316 000. 02 270 000.
B 2.2	Eingabe von Dezimalzahlen im Satz ohne Kennzeichen, Umwandlung in Festkommazahl	7565	02 43 000.
B 2.3	Eingabe einer Dezimalzahl, Umwandlung in tetradische Darstellung (max. 8 Stellen)	7570	02 22 210. 02 14 000.
B 2.3	Eingabe einer Dezimalzahl ohne Kennzeichen (max. 8 Stellen), Umwandlung in Festkommazahl	7571	02 50 000. 02 53 000.
B 4.1	Stanzen eines Wortes in Triaden mit $K_1 = 0$	7563	02 26 000.
B 4.2	Ausgabe einer Gleitkommazahl in Gleitkommadarstellung	7562	02 100 000.
B 4.3	Ausgabe einer Gleitkommazahl in Dezimalschreibweise mit Runden	7561	02 156 000.
B 4.3	Ausgabe einer Gleitkommazahl in Dezimalschreibweise ohne Runden	7560	02 27 210.
B 4.4	Ausgabe von Festkommazahlen in Dezimalschreibweise (Variante 1)	7557	02 12 000.

Baustein	Programm	Ansprungzelle	Inhalt
B 4.5	Ausgabe von Festkommazahlen in Dezimalschreibweise mit Runden (Variante 2)	7556	02 14 000.
B 4.5	Ausgabe von Festkommazahlen in Dezimalschreibweise ohne Runden (Variante 2)	7555	02 27 210.
B 4.5	Ausgabe von Tetraden in Dezimalschreibweise ohne Runden (analog Variante 2)	7554	02 275 000.
B 5.4	Festkommamultiplikation ohne Rückspeicherung	7542	02 132 000.
B 5.4	Festkommamultiplikation mit Rückspeicherung	7543	02 70 000.
B 5.5	Festkommadivision mit Rückspeicherung nach R 1	7541	02 73 000.
B 5.5	Festkommadivision ohne Rückspeicherung	7540	02 217 000.
B 5.2	Gleitkommamultiplikation mit Rückspeicherung nach R 1	7547	02 34 000.
B 5.2	Gleitkommamultiplikation ohne Rückspeicherung	7546	02 34 000.
B 5.3	Gleitkommadivision mit Rückspeicherung nach R 1	7545	02 134 000.
B 5.3	Gleitkommadivision ohne Rückspeicherung	7544	02 134 000.

Zusammenstellung der häufigsten Operationen

Es werden diejenigen verdrahteten Operationen, die bei der praktischen Programmierung am häufigsten benötigt werden, angegeben.

Codierung	Bedeutung (für Adreßteil a)
200 (300)	Konjunktion im Akkumulator: $\langle AC \rangle := \langle AC \rangle \wedge \langle a \rangle$ (mit Vergleichersetzen)
211 (311)	Lesebefehl: $\langle AC \rangle := \langle a \rangle$ (mit Vergleichersetzen)
210 (310)	Addition im Festkomma: $\langle AC \rangle := \langle AC \rangle + \langle a \rangle$ (mit Vergleichersetzen)
221 (321)	Lesebefehl für das Komplement: $\langle AC \rangle := -\langle a \rangle$ (mit Vergleichersetzen)
220 (320)	Subtraktion im Festkomma: $\langle AC \rangle := \langle AC \rangle - \langle a \rangle$ (mit Vergleichersetzen)
330	Vergleichersetzen
331	Vergleichersetzen: $\langle AC \rangle := 0$
240 (340)	Speicherbefehl: $\langle a \rangle := \langle AC \rangle$ (mit Vergleichersetzen)
241	Löschbefehl: $\langle AC \rangle := 0$ $\langle a \rangle := 0$
250 (350)	Konjunktion im Speicher: $\langle a \rangle$ im $\langle AC \rangle \wedge \langle a \rangle$ (mit Vergleichersetzen)
251 (351)	R-Befehl: $\langle AC \rangle := \langle \alpha \rangle$ (mit Vergleichersetzen)
271 (371)	R-Befehl: $\langle AC \rangle := -\langle \alpha \rangle$ (mit Vergleichersetzen)
000	Unbedingter Sprung: > nächster Befehl $\langle := a$
040	Negativsprung: >nächster Befehl $\langle := \begin{cases} a, & \text{falls } \langle AC \rangle < 0 \\ \langle BZ \rangle + 1^0, & \text{falls } \langle AC \rangle \geq 0 \end{cases}$

050 Positivsprung:

$$\text{>nächster Befehl<} := \begin{cases} a, & \text{falls } \langle AC \rangle \geq 0 \\ \langle BZ \rangle + 1^{o}, & \text{falls } \langle AC \rangle < 0 \end{cases}$$

070 Überlaufsprung:

$$\text{>nächster Befehl<} := \begin{cases} a, & \text{falls Überlauf} \\ \langle BZ \rangle + 1^{o}, & \text{falls kein Überlauf} \end{cases}$$

140 Vergleichersprung 1

$$\text{>nächster Befehl<} := \begin{cases} a, & \text{falls Vergleicher 1 oder 3} \\ \langle BZ \rangle + 1^{o}, & \text{falls weder Vergleicher 1 noch 3} \end{cases}$$

150 Vergleichersprung 2

$$\text{>nächster Befehl<} := \begin{cases} a, & \text{falls Vergleicher 2 oder 3} \\ \langle BZ \rangle + 1^{o}, & \text{falls weder Vergleicher 2 noch 3} \end{cases}$$

170 Vergleichersprung 3

$$\text{>nächster Befehl<} := \begin{cases} a, & \text{falls Vergleicher 3} \\ \langle BZ \rangle + 1^{o}, & \text{falls nicht Vergleicher 3} \end{cases}$$

021 Unbedingter Stopp:
< AC > := 0

061 Bedingter Stopp:
< AC > := 0

025 Sprung mit Rückkehrabsicht:
< AC > := < BZ > +1°
> nächster Befehl < := a

030 Ausgabe-Halt
Anwahl bzw. Ausgabe eines Zeichens

K 1	Bedeutung	Eingabebild
0	Maschinenbefehl zur fortlfd. Oktalzahl } Abspeicherung	0 K_2 xxxx $a_1a_2a_3a_4$ BAZ • 0 K_2 t_{10} ... t_0 •
1	Maschinenbefehl zur sofortigen Abarbeitung	1 K_2 xxxx $a_1a_2a_3a_4$ BAZ •
2	Hilfsadresse	2 K_2 xxxx $a_1a_2a_3a_4$ •
3	Dezimalzahl intern in eine GK-Zahl konvertieren	3 (-) d_g ... d_0 •
5	Dezimalzahl in eine FK-Zahl konvertieren , Komma hinten	5 (-) d_g ... d_0 •
B D	} Serviceprogramme	z.B. D 2 $a_1a_2a_3a_4$ • $b_1b_2b_3b_4$ •
ZWR.	Löscht AC-Inhalt vor WM (.)	

K_2	bei Befehlen	bei Hilfsadressen
0	keine Adressenänderung	Relativadresse
1	Adressenänderung mit der 1. Leitadresse	1. Leitadresse = Hauptleitadresse
2 ... 9	Adressenänderung 2 ... 9 Leitadresse	2. Leitadresse ... 9. Leitadresse

K_2	Bedeutung f. D-Programme	Eingabebild
3	Darstellung als Befehl	D 3 $a_1a_2a_3a_4$ • $b_1b_2b_3b_4$ •
4	Befehl bzw. GK-Zahl	dito
5	FK-Zahlen / Befehle	

Befehlsfolge zum Duplizierprogramm (R 300-Code)

```
5000 0200 030
5001 0010 030
5002 0020 030
5003 0230 030
5004 0040 030   Konstanten
5005 0250 030   zur Ausgabe
5006 0260 030   der Zahlen
5007 0070 030
5010 0100 030
5011 0310 030
5012 0320 030
5013 5101 000   Rücksprung
5014 0000 000
5015 5105 000
5016 5107 000
5017 0370 030
5020 0400 030
5021 0610 030
5022 0620 030
5023 0430 030
5024 0640 030
5025 0450 030
5026 0460 030
5027 0670 030
5030 0700 030
5031 0510 030
5032 5111 000
5033 0730 030   Ausgabe-Punkt
5034 5101 000   Rücksprung
5035 0000 000
5036 5113 000
5037 5115 000
5040 5117 000
5041 1210 030
5042 1220 030
5043 1030 030
5044 1240 030
5045 1050 030
5046 1060 030
5047 1270 030
5050 1300 030
5051 1110 030
5052 1120 030
5053 0000 000
5054 0000 000
5055 0000 000
5056 1360 030
5057 5121 000
5060 5123 000
5061 1410 030
5062 1420 030
5063 1630 030
5064 1440 030
5065 1650 030
5066 1660 030
5067 1470 030

5070 1500 030
5071 1710 030
5072 5125 000
5073 5127 000
5074 1740 030
5075 1550 030
5076 5101 000
5077 1410 030
5100 4600 030   <-- Start
5101 5000 011   - Eingabe-Halt-B.
5102 5103 240   - Transport des
5103 0000 000     < AC > nach 5103
5104 5101 000     zur Abarbeitung
5105 0150 030
5106 5101 000
5107 0160 030
5110 5101 000
5111 2720 030
5112 5101 000
5113 0760 030
5114 5101 000
5115 0570 030
5116 5101 000
5117 1000 030
5120 5101 000
5121 1170 030
5122 5101 000
5123 1600 030
5124 5101 000
5125 3520 030
5126 5101 000
5127 1530 030
5130 5101 000
```

Codierung R 300

Lochung 1 2 3 4 5 6 7 8	Eingabe-verschl.	S	Zeichen KB	Zeichen GB	L	Ausgabe-verschl.
	0000		0		L	020X
●	0001		1	<	L	001X
●	0002		2	‖	L	002X
● ●	0003		3	~	L	003X
●	0004		4	[	L	004X
● ●	0005		5	>	L	025X
● ●	0006		6	—	L	026X
● ● ●	0007		7	≈	L	007X
●	0010		8	~	L	010X
● ●	0011		9	]	L	031X
●	0012	S		ZR	L	032X
● ●	0013					013X
●	0014					034X
● ●	0015	S	:	(	L	015X
● ●	0016	S		IS		016X
● ● ●	0017					037X
●	0020		+	=	L	040X
● ●	0021		a	A	L	061X
● ●	0022		b	B	L	062X
● ● ●	0023		c	C	L	043X
● ●	0024		d	D	L	064X
● ●	0025		e	E	L	045X
● ●	0026		f	F	L	046X
● ● ●	0027		g	G	L	067X
● ●	0030		h	H	L	070X
● ●	0031		i	I	L	051X
●	0032	S				052X
● ●	0033 WM	S	.		L	073X
●	0034					054X
● ●	0035	S				075X
● ●	0036	S		IB		076X
● ● ●	0037	S		GB	L	057X
●	0040	S	-	Δ	L	100X
● ●	0041		j	J	L	121X
● ●	0042		k	K	L	122X
● ● ●	0043		l	L	L	103X
● ●	0044		m	M	L	124X
● ●	0045		n	N	L	105X
● ● ●	0046		o	O	L	106X
● ● ● ●	0047		p	P	L	127X

Lochung 1 2 3 4 5 6 7 8	Eingabe-verschl.	S	Zeichen KB	GB	L	Ausgabe-verschl.
○ ○ ○ ● ○ ● ○ ●	0050		q	Q	L	130X
● ○ ○ ● ○ ○ ○ ●	0051		r	R	L	111X
○ ● ○ ● ○ ○ ○ ●	0052		WR			112X
● ● ○ ● ○ ● ○ ●	0053	S				133X
○ ○ ● ● ○ ○ ○ ●	0054					114X
● ○ ● ● ○ ● ○ ●	0055	S				135X
○ ● ● ● ○ ● ○ ●	0056					136X
● ● ● ● ○ ○ ○ ●	0057	S	,	KB	L	117X
○ ○ ○ ○ ● ● ● ●	0060	S		;	L	160X
● ○ ○ ● ○ ○ ● ●	0061		/	)	L	141X
○ ● ○ ● ○ ○ ● ●	0062		s	S	L	142X
● ● ○ ● ○ ● ● ●	0063		t	T	L	163X
○ ○ ● ● ○ ○ ● ●	0064		u	U	L	144X
● ○ ● ● ○ ● ● ●	0065		v	V	L	165X
○ ● ● ● ○ ● ● ●	0066		w	W	L	166X
● ● ● ● ○ ○ ● ●	0067		x	X	L	147X
○ ○ ○ ○ ● ○ ● ●	0070		y	Y	L	150X
● ○ ○ ○ ● ● ● ●	0071		z	Z	L	171X
○ ● ○ ● ● ● ● ●	0072			ZL		172X
● ● ○ ● ○ ● ● ●	0073	S	ß	?	L	153X
○ ○ ● ● ○ ● ● ●	0074		%	*	L	174X
● ○ ● ● ○ ○ ● ●	0075	S	Tab/WM		L	155X
○ ● ● ● ○ ○ ● ●	0076					156X
● ● ● ● ○ ● ● ●	0077	S				177X
○ ○ ○ ○ ○ ○ ○ ○						
○ ● ○ ● ● ● ○ ●	0032	S	WRZL/SM		L	272X
○ ● ○ ● ○ ○ ● ○	0072	S	⊕	⊖		352X
● ● ● ● ● ● ● ●	-	-	Irrung		L	377X

S Zeichen stellt Eingabehaltbefehl in Sprung um

L Zeichen kann mit der SE 5L 08008 gelocht werden

Literaturnachweis

1. Programmierung und Bedienung des digitalen Kleinrechners C 8205, Heft 2 und Heft 4
   VEB Kombinat Zentronik, Rechenelektronik
   Meiningen/Zella-Mehlis

2. Programmiersystem PROSA
   Cellatron C 8205
   VEB Kombinat Robotron